

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Aplicação de Técnicas de Realidade Virtual na Simulação do Parto

Sérgio Luís Abrantes Pinto

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Renato Manuel Natal Jorge

Coorientador: Marco Paulo Lages Parente

25 de Junho de 2017

Resumo

A dissertação “Aplicação de técnicas de realidade virtual na simulação do parto” tem como objetivo a criação de uma aplicação em realidade virtual, partindo do modelo biomecânico utilizado na análise do parto humano, através do canal vaginal.

Este projeto surge da necessidade de desenvolver um método imersivo e interativo, utilizando diversas ferramentas de realidade virtual, como o Oculus Rift e o dispositivo Leap Motion, que permita a visualização tridimensional de modelos simulados na área da biomecânica, sendo que o caso da presente dissertação o referente ao parto vaginal de um ser humano.

Nas ferramentas utilizadas ao longo deste projeto inclui-se *software*, como o Abaqus (ferramenta utilizada na simulação numérica), Blender e Unity (usado para a criação de ambientes virtuais, quer a nível de videojogos como em aplicações científicas). Além disso, irão também ser focados os dispositivos de *hardware* relativos à realidade virtual. Nestes últimos insere-se o Oculus Rift, aparelho acoplado à cabeça do utilizador, que proporciona a imersão num mundo virtual, estimulando os sentidos do corpo humano, e o dispositivo Leap Motion, capaz de detetar os vários segmentos da mão do utilizador (palma, dedos, entre outros) e também alguns gestos que realizam alguma função no modelo simulado ou no mundo virtual, personalizável através de programação antes da criação da aplicação.

De modo a realizar a aplicação referida, foram abordados vários tópicos de estudo, entre os quais a anatomia da pelve humana feminina, e analisados estudos biomecânicos relativos ao parto. Foram ainda investigadas as ferramentas atualmente utilizadas na área da realidade virtual.

Por fim, após a descrição de todas as etapas seguidas, concluiu-se o projeto, tendo-se obtido a aplicação desejada, mas ainda com algum trabalho futuro que servirá para refinar as técnicas utilizadas.

Abstract

This thesis, intitled “Application of virtual reality techniques to a birth simulation”, intends to create an application using virtual reality, starting with a biomechanical model related to the analysis of the human birth through the vaginal canal.

This project arises from the need to develop an immersive and interactive method, using several virtual reality tools, such as the Oculus Rift or the Leap Motion device, which allows a tridimensional visualization of the models simulated in the biomechanical area. The model used in this thesis portraits the human birth through the vaginal canal.

In the tools utilized throughout the development of this project, several software is included, such as Abaqus (a tool used in numerical simulation), Blender and Unity (used to create virtual environments, either applied to videogames or to scientific apps). Other than that, some hardware regarding the virtual reality will be described. In the last ones referred, the Oculus Rift is one of the most known, and consists of a head-mounted device, allowing the user to be immersed in the virtual world, stimulating the human senses. Another hardware is the Leap Motion device, which is able to detect the segments of the user’s hands (such as the palm of the hand, each finger, ...), and also allows to associate a gesture to some functions, customizable through programming before the creation of the application.

In order to create the referred application, several topics were studied, including the anatomy of the human pelvis, with emphasis in the female case, and some biomechanical studies were analyzed. The currently used tools in the field of virtual reality were also investigated.

Lastly, after the description of all the steps, the project was completed, and the product obtained was the desired application. Despite this, there is still some future work to be done, which will result in the refinement of the used techniques.

Agradecimentos

Em primeiro lugar, gostaria de agradecer ao meu orientador, o Prof. Renato Natal Jorge, e ao meu coorientador, o Prof. Marco Parente, por todo o apoio dado, e por me terem guiado ao longo dos últimos meses, principalmente nos momentos em que achava que o caminho para avançar era muito mais comprido do que aquele por onde me encaminhavam.

Aos meus colegas de laboratório quero agradecer o fantástico ambiente que me proporcionaram durante o desenvolvimento desta dissertação, pela ajuda na resolução mais prática daqueles assuntos onde a minha experiência era mais reduzida e pelo sentimento de camaradagem que me fizeram sentir desde o primeiro dia de trabalho.

Gostaria também de agradecer ao Projeto NORTE-01-0145-FEDER-000022 – SciTech – Science and Technology for Competitive and Sustainable Industries , cofinanciado pelo Programa Operacional Regional do Norte (NORTE2020), através do Fundo Europeu de Desenvolvimento Regional (FEDER) pela possibilidade de concretização do projeto.

Aos meus amigos do dia-a-dia quero dizer um muito sincero obrigado por me fazerem sentir em casa nos últimos anos, e por me ajudarem a abstrair do facto de que a minha primeira casa fica a tantas e tantas horas de distância do Porto, e por me terem acompanhado neste percurso, dando-me inúmeros momentos que não esquecerei.

Quero dar um especial obrigado à Tuna de Engenharia da Universidade do Porto, instituição que me acolheu desde as primeiras semanas nesta cidade na altura nova e desconhecida para mim, e que me tornou no homem que eu sou hoje. Um grande agradecimento por todas as competências que nunca de outra forma teria adquirido, e que complementaram a minha formação, fornecendo-me características que não fazem parte do plano de estudos. Acrescento ainda que estou eternamente grato pela segunda família que me deu, onde encontrei amigos para a vida (não vou citar nomes, pois vocês sabem quem são).

Por último, mas não menos importante (antes pelo contrário), quero agradecer aos meus pais, pela paciência que tiveram comigo nos últimos anos, pelas dificuldades a que se submeteram para que eu pudesse ter uma vida em que nada me faltasse numa cidade onde me encontrava "sozinho", pelas poucas visitas que podíamos ter durante todo o ano, por me terem permitido frequentar um curso superior quando se contam pelos dedos das mãos aqueles que tiveram esse privilégio na minha família, e por me terem feito ser a pessoa que eu sou hoje.

Sérgio Pinto

*“It’s ok if you fall down and lose your spark.
Just make sure that when you get back up,
you rise as the whole damn fire.”*

Colette Werden

Índice

1	Introdução	1
1.1	Enquadramento	1
1.2	Motivação	1
1.3	Objetivos	2
1.4	Estrutura da Dissertação	2
2	Revisão Bibliográfica	3
2.1	Estrutura Anatômica da Pelve Feminina	3
2.1.1	Órgãos da Cavidade Pélvica	4
2.1.2	Pavimento Pélvico	6
2.2	Realidade Virtual	8
2.2.1	Tecnologias Atuais	9
3	Ferramentas Utilizadas	13
3.1	<i>Software</i>	13
3.1.1	Abaqus	13
3.1.2	Ficheiros STL (Estereolitografia)	14
3.1.3	<i>Virtual Reality Modeling Language</i>	15
3.1.4	Blender	16
3.1.5	Ficheiros FBX	17
3.1.6	Unity	18
3.2	<i>Hardware</i>	19
3.2.1	Oculus Rift	19
3.2.2	Leap Motion	21
4	Biomecânica do Parto	23
4.1	Objetivos do Estudo Biomecânico do Parto	23
4.2	Evolução do Estudo do Parto	24
4.3	Análise Biomecânica do Parto	25
4.4	Disfunções Resultantes do Parto e Conclusões Obtidas pelos Estudos	27
4.5	Modelo Biomecânico Utilizado	29
5	Importação do Modelo Biomecânico para a Realidade Virtual	33
5.1	Conversão para o Formato VRML	33
5.2	Conversão para o Formato FBX	39
5.3	Construção da Aplicação no Unity - Importação do Modelo para Realidade Virtual	41

6	Interface com o Utilizador	49
6.1	Introdução do Dispositivo Leap Motion	49
6.2	Determinação das Funcionalidades a Implementar	51
6.3	Implementação das Funcionalidades	52
6.4	Criação do Executável	58
7	Conclusões e Trabalho Futuro	61
7.1	Conclusões	61
7.2	Trabalho Futuro	62
A	Ficheiro de Código em C# - FrameByFrameAnimation	65
B	Ficheiro de Código em C# - ModelRotation	69
	Referências	73

Lista de Figuras

2.1	Representação da Pelve Óssea (adaptado de [4])	4
2.2	Órgãos da Cavidade Pélvica (adaptado de [7])	5
2.3	Músculos e Tecidos Conjuntivos Pélvicos (adaptado de [8])	7
2.4	Arquitetura básica de um sistema de realidade virtual e terminologias associadas (retirado de [9])	8
2.5	Exemplos de Dispositivos Visuais	10
2.6	Luva de Dados (retirado de [15])	11
3.1	Exemplo de simulação de deformação de um objeto usando Abaqus (adaptado de [17])	14
3.2	Redundância de vértices em STL. Na imagem da esquerda encontra-se representada a forma como uma superfície se encontra descrita num ficheiro STL, havendo vértices duplicados ao longo do conjunto de dados. Na imagem da direita encontra-se demonstrado como uma superfície deveria estar descrita para evitar redundâncias. (retirado de [18])	15
3.3	Tipos de formatos suportados pelo Blender (adaptado de [25])	17
3.4	Interface do Blender	17
3.5	Estrutura (em árvore) de um ficheiro FBX (retirado de [28])	18
3.6	Interface do Unity	19
3.7	Hardware - Oculus Rift e Controladores Touch	20
3.8	Leap Motion	21
3.9	Gestos reconhecidos pelas funções fornecidas na API do Leap Motion (retirado de [39])	22
4.1	Modelo do crânio do feto. A branco encontra-se representada a superfície óssea, e a preto as regiões correspondentes às fontanelas e suturas. (retirado de [40])	26
4.2	Representação dos pontos de referência do feto, para controlo do seu movimento vertical (retirado de [40])	27
4.3	Simulação da episiotomia. À esquerda a incisão foi realizada a 60° do eixo vertical, no sentido positivo, e com comprimento de 30 mm. À direita a orientação foi de 45° e o comprimento do corte foi de 10 mm. (adaptado de [40]).	28
4.4	<i>Frames</i> utilizados na simulação do parto incluindo a análise do deslocamento (U) do feto (modelo utilizado na aplicação). Na última imagem é possível observar o valor dos deslocamentos mínimo (0,0 mm) e máximo (199,9 mm) do feto.	31
4.5	<i>Frames</i> utilizados na simulação do parto em relação ao deslocamento (U) dos MPP (feto não considerado na análise). Na última imagem é possível observar os valores do deslocamento mínimo (0,0 mm) e máximo (44,54 mm) dos MPP.	32

5.1	Modelo biomecânico no ambiente do <i>software</i> Abaqus	34
5.2	Janela de seleção da variável de análise no <i>software</i> Abaqus	34
5.3	Seleção dos componentes dos MPP	35
5.4	<i>Frames</i> exportados para o formato VRML correspondentes aos MPP, começando no <i>frame</i> 0 do <i>step</i> 0 (canto superior esquerdo) e terminando nos <i>frames</i> 0 e 803 do <i>step</i> 7 (imagens da última linha, correspondentes ao primeiro e último <i>frames</i> do último <i>step</i>).	36
5.5	Janela de seleção de componentes a incluir no modelo (evidenciados encontram-se os elementos correspondentes ao feto)	37
5.6	<i>Frames</i> exportados para o formato VRML correspondentes ao feto, começando no <i>frame</i> 0 do <i>step</i> 0 (canto superior esquerdo) e terminando nos <i>frames</i> 0 e 803 do <i>step</i> 7 (imagens da última linha, correspondentes ao primeiro e último <i>frames</i> do último <i>step</i>).	38
5.7	Modelo com elementos desnecessários. Imediatamente abaixo do corpo do feto podem-se observar alguns componentes da escala do modelo e alguns elementos de texto.	39
5.8	Janela de seleção de elementos extra do modelo. Esta área encontra-se em “Viewport » Viewport Annotation Options...”	40
5.9	Objeto completamente processado e pronto para exportação para o formato FBX.	40
5.10	Características do ponto de luz inserido no Unity	41
5.11	Sistema de coordenadas do Unity	42
5.12	Sistema esquelético da pelve inserido no Unity	42
5.13	Posição, orientação e escalamento do sistema esquelético no Unity (<i>GameObject</i> com o nome “bones”)	43
5.14	<i>GameObject</i> “Shape_IndexedFaceSet.013”	44
5.15	Seleção dos <i>GameObjects</i> , demonstrando a hierarquia dos mesmos. É também possível visualizar o resultado da calibração entre o sistema esquelético e um dos <i>frames</i> do modelo biomecânico.	45
5.16	Exemplo do modelo biomecânico inserido no ambiente virtual. No caso retratado observa-se o <i>frame</i> 0 do <i>step</i> 0 da simulação.	46
5.17	Ativação ou desativação de um <i>GameObject</i> no Unity. Caso a caixa existente no lado esquerdo do nome do objeto possua um visto, o <i>GameObject</i> encontra-se ativo.	46
5.18	Hierarquia final do <i>GameObject</i> “bones”. Os objetos “FEMAP_0” e “FEMAP_1” correspondem ao sistema esquelético da pelve. O objeto “AnimationFrames” possui associado a si o código em C# referido e agrega os <i>frames</i> do modelo biomecânico, para simplificar a modificação do mesmo. Os objetos “m_step_X” (com X entre 0 e 7) correspondem aos 8 <i>frames</i> dos MPP, enquanto que os objetos “stepY” (com Y entre 0 e 7) correspondem ao feto.	47
5.19	Modelo biomecânico final inserido no Unity, com os oito <i>frames</i> dos MPP e do feto	48
6.1	Volume dentro do qual o Leap Motion consegue detetar objetos. Na imagem é possível observar que os eixos pré-definidos pelo Leap Motion não correspondem aos eixos do Unity, pelo que é necessário um ajuste a nível do código. (retirado de [34])	50
6.2	Hierarquia do <i>GameObject</i> “LMHeadMountedRig”, contendo o objeto responsável pela definição do volume de deteção do Leap Motion, “LeapHandController”, e os objetos correspondentes às mãos e dedos virtuais.	50
6.3	Modelo “RigidRoundHand” para as mãos utilizado na simulação	51
6.4	Evolução do desenvolvimento do menu na simulação	52

6.5	Botões e <i>sliders</i> do menu. 1- Escolha do <i>frame</i> quando animação está pausada; 2- Retroceder um <i>frame</i> quando animação está pausada; 3- Play/Pausa da animação; 4- Avançar um <i>frame</i> quando animação está pausada; 5- Rotação do modelo nos eixos X e Y; 6- Escolha da velocidade da animação.	53
6.6	Descrição do <i>GameObject</i> “bones”, com a associação do <i>script</i> “ModelRotation” e da caixa de colisão.	54
6.7	Movimentos de supinação e pronação do pulso. O S na figura corresponde ao movimento de supinação e o P ao movimento de pronação. (retirado de [71]) . .	56
6.8	Implementação da funcionalidade de ativação/desativação do menu. No caso retratado, a mão que controla a função é a mão esquerda.	57
6.9	Caixa de colisão (a verde) associada ao <i>GameObject</i> “bones”	58
6.10	Janela de exportação da cena virtual para os vários ambientes permitidos	59

Lista de Tabelas

3.1	Comparação das Características dos DK 1 e 2 dos Oculus Rift (adaptado de [33])	20
3.2	Funções disponibilizadas na API do Leap Motion [37]	22

Abreviaturas e Símbolos

cm	Centímetro(s)
mm	Milímetro(s)
HMD	<i>Head-Mounted Display(s)</i>
HCD	<i>Head-Coupled Display(s)</i>
VRML	<i>Virtual Reality Modeling Language</i>
HTML	<i>HyperText Markup Language</i>
WWW	<i>World Wide Web</i>
DK	<i>Development Kit(s)</i>
OLED	<i>Organic Light-Emitting Diode(s)</i>
LED	<i>Light Emmiting Diode(s)</i>
SDK	<i>Software Development Kit</i>
API	<i>Application Programmer Interface</i>
N	Newton(s)
fps	<i>Frames por segundo</i>
STL	<i>Stereolithography</i>
CAD	<i>Computer Aided Design</i>
ASCII	<i>American Standard Code for Information Interchange</i>
OpenGL	<i>Open Graphics Library</i>
SO	Sistema Operativo
ODB	<i>OpenDocument Database</i>

Capítulo 1

Introdução

1.1 Enquadramento

O parto é um processo fisiológico natural que possui um impacto bastante significativo na anatomia da mulher, principalmente a nível muscular, e do próprio feto.

De modo a melhor entender este processo foram realizados vários estudos que resultaram em conjuntos de dados referentes a diversas patologias e anormalidades, e que demonstram que o parto (particularmente o primeiro) tem a contribuição mais significativa no aparecimento e desenvolvimento de inúmeras disfunções do pavimento pélvico e de lesões no feto.

Ao longo dos últimos anos tem sido criado um modelo computacional tridimensional que permite o estudo da complexa anatomia da região pélvica do ser humano do género feminino, bem como dos esforços envolvidos durante o parto, mas que apenas permite a visualização em *software* específico para o tipo de ficheiro desse modelo.

Utilizando técnicas de simulação numérica, como o método dos elementos finitos, e recorrendo a modelos biomecânicos, tem sido possível a determinação dos níveis de deformação a que se sujeitam os diferentes órgãos e tecidos, assim como o resultado dos vários parâmetros que podem afetar as anormalidades subsequentes ao parto.

1.2 Motivação

O presente projeto surge da necessidade de se associar o modelo biomecânico atualmente existente (resultante de simulações numéricas que aplicam o método dos elementos finitos) e técnicas de realidade virtual.

Com esta associação, procura-se a criação de um sistema imersivo em ambientes virtuais, aplicado na área da saúde/cirurgia, que proporcione uma visualização tridimensional das simulações biomecânicas.

Espera-se que no futuro se possa obter uma ferramenta de treino para profissionais da área médica, mantendo-se as propriedades relevantes que tornam esta simulação num caso de estudo.

1.3 Objetivos

A presente dissertação tem como objetivo principal o desenvolvimento de uma aplicação, com recurso a ferramentas de realidade virtual, que associa o modelo relativo ao parto do ser humano (obtido através de análise de simulação biomecânica), proporcionando uma visualização tridimensional com possibilidade de interação entre o utilizador e o modelo simulado.

Funciona assim como um pós-processamento ao método dos elementos finitos.

1.4 Estrutura da Dissertação

A presente dissertação encontra-se dividida em seis capítulos relativos ao desenvolvimento do projeto.

No capítulo 2 pode-se encontrar a revisão bibliográfica, que contém todas as definições e informações teóricas que serão necessárias para o completo entendimento do trabalho desenvolvido ao longo dos últimos meses. Nele existem dois grandes tópicos principais: o primeiro relativo à anatomia da pelve humana, particularmente no caso da mulher, e o segundo alusivo à realidade virtual, no que diz respeito à sua definição e às ferramentas que existem atualmente.

No capítulo 3 poderão encontrar-se as ferramentas de trabalho utilizadas ao longo do projeto, tanto a nível de *software* como a nível de *hardware*. Serão abordados em particular os Oculus Rift, o Leap Motion e o Unity, devido à sua maior importância na concretização dos objetivos da presente dissertação.

No capítulo 4 poderão encontrar-se todas as informações relativas ao modelo biomecânico do parto humano através do canal vaginal, desde quais os métodos utilizados ao longo do tempo para análise do parto até aos resultados obtidos recentemente com o estudo do modelo utilizado nesta dissertação.

No capítulo 5 tratar-se-á de todos os processos que foram utilizados durante o desenvolvimento da aplicação, em particular das conversões entre formatos de ficheiro necessárias para transformar o modelo existente inicialmente em formatos compatíveis com realidade virtual. Será também abordado o processo utilizado para a construção da aplicação em si.

No capítulo 6 encontram-se descritas todas as formas de interação que existem entre o utilizador e o modelo biomecânico já em realidade virtual, incluindo a execução de diversas funcionalidades através do teclado do computador e do dispositivo Leap Motion. Contempla ainda a criação do ficheiro executável, dando a possibilidade à aplicação de funcionamento em qualquer computador que suporte realidade virtual.

Por fim, no capítulo 7 são descritas as conclusões a que se chegou com o desenvolvimento do projeto. Pode ser encontrada ainda a listagem dos problemas encontrados durante o período de trabalho e quais os tópicos a serem acrescentados no futuro.

Capítulo 2

Revisão Bibliográfica

Para o desenvolvimento da aplicação em realidade virtual contendo as simulações biomecânicas, cujo foco se encontra na região pélvica da mulher, é necessário, antes de tudo, definir e investigar mais sobre cada uma destas vertentes.

Desta forma, nos subcapítulos seguintes irão ser referidos vários tópicos com relevância para atingir os objetivos propostos. No subcapítulo 2.1 irá ser apresentada a anatomia da região pélvica da mulher, incluindo os músculos e estrutura óssea existentes. O subcapítulo 2.2 referir-se-á à realidade virtual, e incluirá qual o *hardware* e *software* existentes no mercado atual.

2.1 Estrutura Anatômica da Pelve Feminina

O desenvolvimento das aplicações em realidade virtual usarão modelos contendo a estrutura corporal designada de pelve, e, portanto, para que a compreensão dos mesmos seja possível, é necessário entender quais os constituintes da anatomia pélvica e quais as funções de cada estrutura.

A cavidade pélvica, formada pelo conjunto dos ossos, órgãos, músculos e ligamentos desta zona, possui várias funcionalidades, nomeadamente proteger órgãos e promover a sua sustentação [1]. O seu sistema esquelético é responsável por grande parte destas funções, além de transmitir o peso do tronco e membros superiores para os membros inferiores. No caso da mulher, contribui para a proteção do feto em desenvolvimento e conduz o mesmo ao exterior durante o parto [2].

O sistema esquelético da cavidade pélvica é constituído por três ossos principais: o sacro, o cóccix e o ilíaco (que por sua vez se divide em três porções - o ílio, o ísquio e a púbis) [figura 2.1]. No interior da cavidade formada por estes ossos, encontram-se alojados alguns órgãos, como a bexiga, a vagina, o útero e o reto.

A organização estrutural e funcional destes órgãos é garantida pela ação dos músculos e ligamentos que se encontram inseridos nos ossos da pelve e, desta forma, torna-se essencial que estes se mantenham íntegros, para que possam realizar a sua ação da melhor forma e não ocorra nenhuma disfunção de sistemas [3].

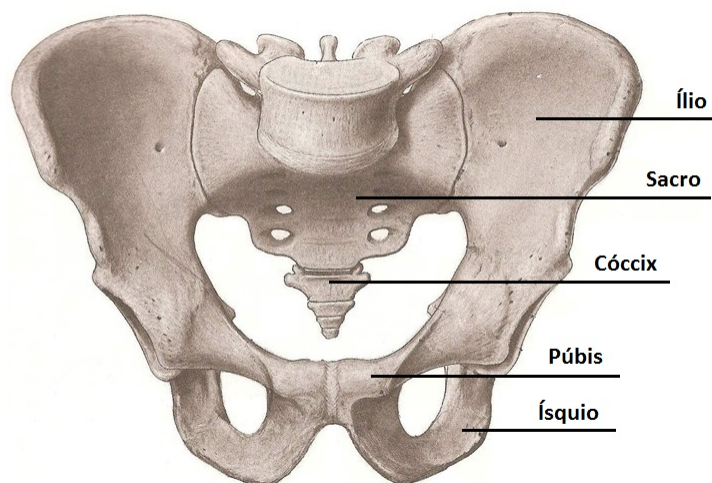


Figura 2.1: Representação da Pelve Óssea (adaptado de [4])

2.1.1 Órgãos da Cavidade Pélvica

Bexiga

A bexiga é um órgão pertencente ao aparelho urinário do ser humano, que se situa na parte inferior abdominal do corpo. Consiste essencialmente num reservatório muscular oco, inserido na cavidade pélvica, localizado posteriormente à sínfise púbica e anteriormente ao colo do útero e vagina.

A sua função essencial é a de armazenamento de urina, filtrada pelos rins, e transportada pelos ureteres, para posterior excreção. Para melhor cumprir esta funcionalidade, as suas paredes possuem fibras musculares, dispostas de forma a que a distensão da bexiga seja possível de forma concêntrica, o que provoca um aumento do volume interior do reservatório [2, 5].

Uretra

A uretra é um órgão tubular, que conduz a urina da bexiga até ao exterior do corpo.

A uretra feminina é significativamente mais curta que a masculina, e o controlo voluntário da continência urinária é realizado por um músculo uretral exterior à uretra (esfíncter), através da sua constrição [2, 5].

Vagina

A vagina é um canal que possui aproximadamente 10 cm de comprimento, mas que pode variar consideravelmente. Estende-se desde o colo do útero até a um orifício localizado no vestíbulo da

vulva, e que contacta com o exterior do corpo.

A sua parede anterior encontra-se adjacente à bexiga, enquanto que a sua parede posterior está adjacente ao reto. Estas paredes são constituídas por três camadas: uma camada muscular mais exterior, uma mucosa e uma camada adventícia.

A camada mais interna das paredes é formada por fibras musculares longitudinais que têm a capacidade de distensão, possibilitando assim o alargamento do canal vaginal durante a relação sexual e no momento de descida fetal no parto [2, 5].

Reto

O reto é uma estrutura muscular, constituinte do tubo digestivo, que atravessa a pelve e se estende até ao canal anal, relacionando-se com o exterior através do ânus.

Encontra-se situado numa zona com importantes estruturas musculares que regulam a defecação, como o esfíncter anal interno e externo [2, 5].

Útero

O útero é um órgão muscular, oco e que cujas paredes possuem uma espessura bastante considerável. Localiza-se na zona inferior da pelve, anterior à bexiga e posterior ao reto [6].

É neste órgão que o feto se desenvolve durante os nove meses de gestação, pelo que a sua integridade é essencial, caso contrário várias dificuldades podem surgir durante uma gravidez.

Durante o período de desenvolvimento fetal, o útero sofre alterações a nível anatómico (é uma estrutura que se adapta ao tamanho do feto) e fisiológico, de modo a acomodar o embrião e a assegurar as suas necessidades [6].

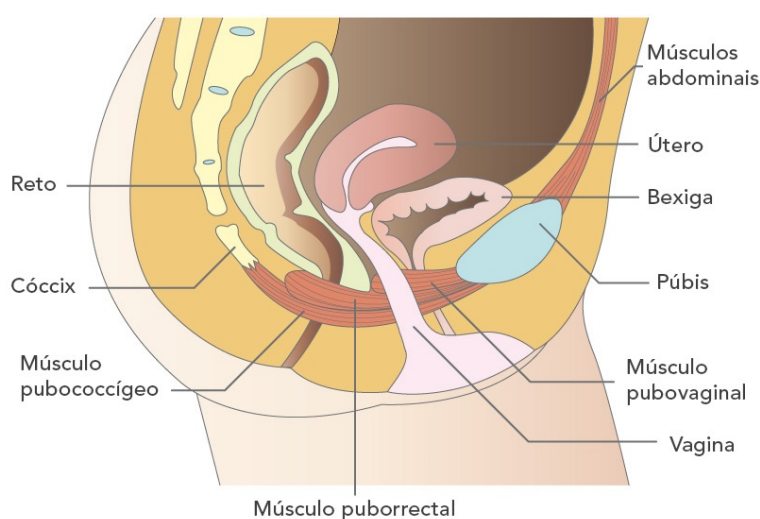


Figura 2.2: Órgãos da Cavidade Pélvica (adaptado de [7])

2.1.2 Pavimento Pélvico

O pavimento pélvico é um conjunto de estruturas musculares e ligamentosas que encerram a cavidade pélvica na sua parte inferior (ligando a púbis anteriormente e o sacro posteriormente).

Possui várias funções, sendo uma das principais o suporte que fornece aos órgãos pélvicos, contrariando a força da gravidade e mantendo os órgãos da cavidade pélvica na sua posição natural.

Devido a este facto, a manutenção da integridade do pavimento torna-se essencial. As diferentes estruturas constituintes deste estão sob constante pressão e forças que contribuem para a disrupção do seu normal funcionamento. Vários fatores que aumentam as forças exercidas ou diminuem o limiar de elasticidade das diferentes estruturas podem resultar numa lesão que terá impacto ao nível funcional, e que poderão levar ao aparecimento de disfunções do pavimento pélvico (por exemplo, incontinência urinária, incontinência fecal ou prolapso de órgãos pélvicos). Entre os vários fatores de risco que contribuem para estas disfunções destacam-se a predisposição genética, obesidade, envelhecimento, gravidez e o parto vaginal [2, 5].

2.1.2.1 Músculos do Pavimento Pélvico (MPP)

Existem duas grandes estruturas no pavimento pélvico: o diafragma pélvico e a membrana perineal, estando a primeira localizada na zona mais inferior e central da pelve e a segunda na zona mais superficial e periférica do pavimento pélvico.

De seguida irão ser descritas algumas características destas estruturas.

Diafragma Pélvico

Esta estrutura muscular ocupa a maior parte do pavimento pélvico e é formado por dois músculos principais: o coccígeo e o músculo elevador do ânus, sendo o segundo o maior constituinte do pavimento.

O músculo elevador do ânus ocupa quase inteiramente a pelve e trata-se de uma estrutura com pouca espessura e que se encontra inserido na superfície interior da pelve (possui, em particular, inserções na sínfise púbica e no ísquio) [6]. As suas principais funções são a de suporte dos órgãos pélvicos na sua posição normal, a promoção da resistência dos mesmos em relação ao aumento da pressão intra-abdominal e o direcionamento dos órgãos responsáveis pela defecação e secreção de urina para a púbis, auxiliando na continência fecal e urinária através da sua contração.

O coccígeo é um músculo que se situa superiormente ao elevador do ânus, e que participa na flexão do cóccix após a defecação ou parto. Também ajuda na sustentação dos órgãos pélvicos, em conjunto com o elevador do ânus [2, 5].

Membrana Perineal

A membrana perineal, responsável pela ação voluntária da continência, é uma estrutura que funciona como ponte entre a púbis e o corpo perineal. Esta atua na sustentação da vagina e da uretra contribuindo para a continência e atuando como esfíncter destas regiões [5].

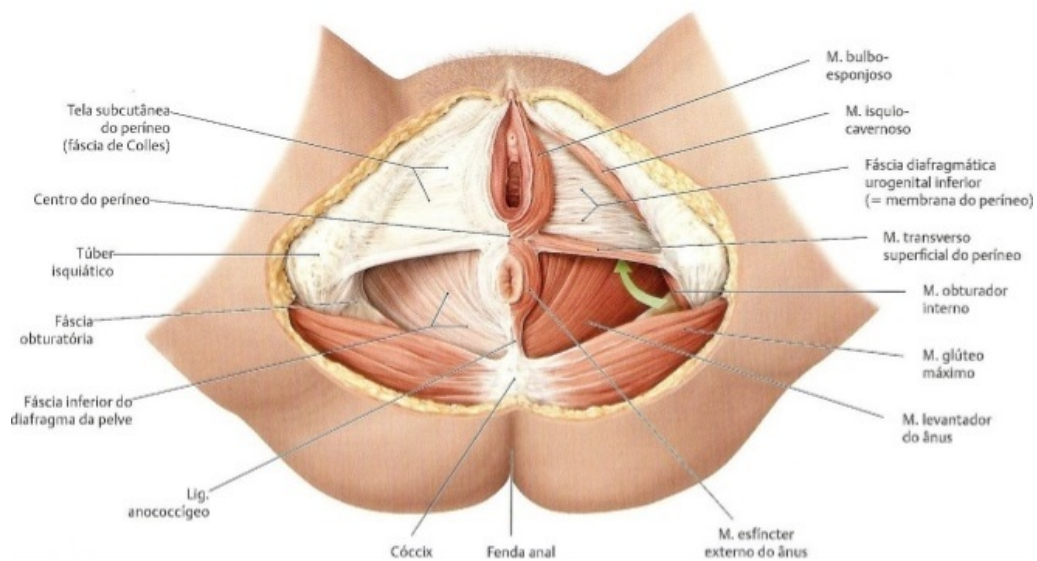


Figura 2.3: Músculos e Tecidos Conjuntivos Pélvicos (adaptado de [8])

2.1.2.2 Estruturas Ligamentosas da Pelve

No pavimento pélvico estas são compreendidas pelo tecido conjuntivo que compõe o complexo de fáscias (tecidos fibrosos que envolvem e separam cada músculo ou grupo muscular, de modo a manter as suas células juntas, e ligando-os aos tendões ou inserindo-os nos ossos [2]) e ligamentos que têm como objetivo o fortalecimento e inserção dos órgãos, protegendo-os das constantes alterações de forças.

Estes tecidos encontram-se classificados em três níveis de suporte distintos, relacionados com a superficialidade das estruturas que suportam, o que permite identificar os sintomas das disfunções pélvicas (causadas maioritariamente pela gravidez, parto ou envelhecimento, fenómenos que comprometem as características funcionais dos seus constituintes), uma vez que cada perturbação num nível pode resultar em prolapsos de diferentes estruturas de órgãos [5].

2.1.2.3 Articulações do Pavimento Pélvico

Além de todas as estruturas já referidas, existe ainda uma, classificada como articulação, e que desempenha uma função importante ao longo do trabalho de parto. Como a maioria das articulações, a sínfise púbica é responsável pela ligação de outras estruturas, nomeadamente das duas extremidades da púbis. É constituída principalmente por tecido de ligação cartilaginoso [6].

A sínfise púbica permite que existam pequenos ângulos de rotação, e até algum deslocamento dos ossos, a nível do sacro e ílio. Em fases avançadas da gestação e durante o parto é possível que ocorra alguma disjunção entre os ossos púbicos [6], facilitando a abertura do pavimento pélvico durante a descida fetal.

2.2 Realidade Virtual

A realidade virtual é um campo de estudos que permite a criação de um ambiente totalmente sintético, gerado com o auxílio de um computador, que fornece uma experiência de completa imersão do utilizador utilizando técnicas de computação gráfica.

Trata-se de uma experiência sintética no sentido em que a estimulação sensorial (possível em todos os sentidos do corpo humano) é criada, simulada e controlada pelo sistema (ver figura 2.4). Este é constituído por um conjunto de dispositivos que fornecem as saídas na forma de estímulos, *outputs*, por sensores que detetam as ações do utilizador e por um computador que processa essa ação e gera o *output* no *display* (que pode não ser apenas visual, mas também sob a forma de qualquer outro tipo de dispositivo sensorial) [9].

O ambiente onde os mundos são criados possuem características tridimensionais e, portanto, permite que exista a sensação de profundidade e de relação espacial entre objetos, incluindo também características do mundo real.

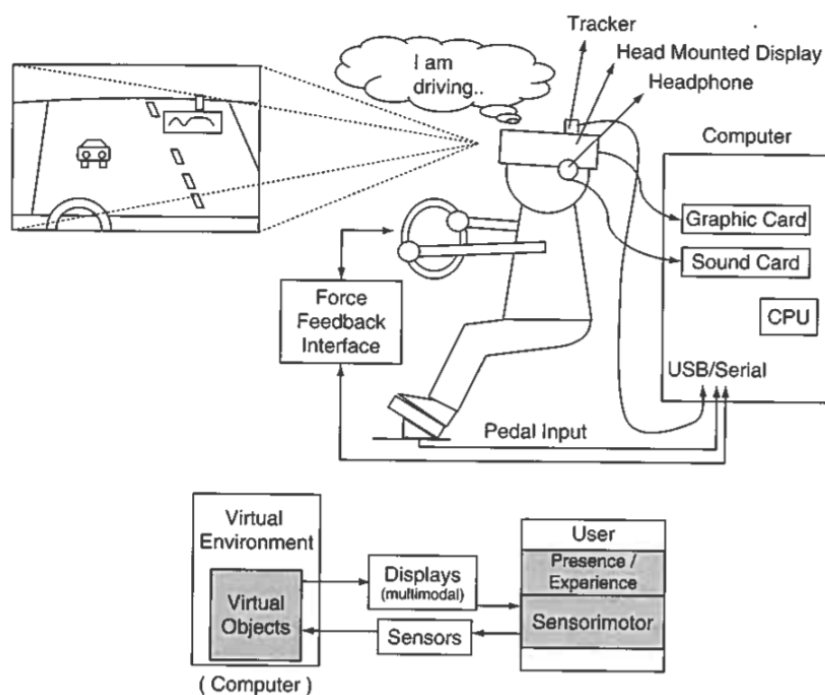


Figura 2.4: Arquitetura básica de um sistema de realidade virtual e terminologias associadas (retirado de [9])

Os objetivos principais da utilização da realidade virtual incluem o treino de profissionais, educação com uso de casos mais práticos e o entretenimento, permitindo aos utilizadores uma experiência de algo que de outra forma seria muito complicada ou impossível de atingir na vida real.

Existem dois pilares principais que caracterizam a realidade virtual: o primeiro, intitulado de

"presença", refere-se ao grau em que os utilizadores sentem que se encontram noutra localização que não aquela em que realmente estão, de modo a criar uma experiência virtual vívida. A presença é conseguida através da fidelidade dos sensores, que permitem a melhor expressão da realidade, ou através do grau de interatividade (se o ambiente permitir a manipulação de objetos, a navegação do utilizador no meio envolvente, a simulação de movimentos e deformações, entre outras funcionalidades). O segundo refere-se à interação tridimensional em múltiplos "canais", ou seja, à inclusão de vários sentidos humanos, envolvendo assim o corpo inteiro do utilizador [9].

Uma vez que o mundo criado responde aos estímulos do utilizador (*input* do sistema), e se adapta a eles, o que faz dele não estático, pode considerar-se que existe interatividade em tempo real na realidade virtual [10].

É bastante comum que a realidade virtual seja confundida com realidade aumentada, mas de facto são bastante diferentes entre si, uma vez que a segunda é uma combinação de uma cena real, vista por um utilizador, com uma cena virtual, criada por um computador, que acrescenta e “aumenta” a cena real.

2.2.1 Tecnologias Atuais

Para que o utilizador se sinta verdadeiramente imerso e possa interagir com o ambiente virtual, existem diversos dispositivos computacionais para a entrada e saída de dados, que procuram estimular a maior quantidade de sentidos e obter eficientemente os movimentos do utilizador.

2.2.1.1 Dispositivos de Saída de Dados

Tratam-se daqueles dispositivos que mostram o ambiente e o resultado da interação entre o utilizador e o meio criado pelo sistema. O foco destes aparelhos é o sentido da visão, procurando isolar o usuário do meio que o envolve. O principal problema encontrado pelos sistemas deste tipo é a estereoscopia, presente no ser humano e no modo como ele observa o mundo real, e lida com a percepção de profundidade através das imagens individuais captadas por cada olho [11].

Dispositivos Visuais

Existem dois tipos de dispositivos adaptados a sistemas óticos: vídeo-capacetes e *displays* acoplados à cabeça, e monitores e sistemas de projeção.

O primeiro tipo encontra-se dividido em vídeo-capacetes (ou *Head-Mounted Displays* - HMD) e em *Head-Coupled Displays* (HCD ou dispositivos acoplados à cabeça) (ver figura 2.5 (a)). Os primeiros distinguem-se por possuírem sensores para detetar movimentos através da variação da orientação e posição da cabeça. São aqueles que mais isolam o utilizador do mundo real, e são compostos por dois pequenos ecrãs e um conjunto de lentes especiais que ajudam a focalizar as imagens e a ampliar o campo visual do mundo virtual. Podem, no entanto, permitir a visualização do mundo real, refletindo o ambiente virtual nas lentes, em frente dos olhos. Os dispositivos mais conhecidos atualmente são os Oculus Rift e os Google Glass. Os segundos funcionam através de um *display* montado sobre um braço mecânico, o que permite uma rápida e fácil transição entre

o ambiente virtual e a interação com o monitor, teclado ou dispositivo que permita o controlo da simulação, controlo esse que é necessário através de outro dispositivo de entrada.

Os monitores e sistemas de projeção implicam que seja necessário estar constantemente a olhar para o ecrã (figura 2.5 (b)). O utilizador necessita de um dispositivo de entrada de dados para controlar a movimentação no ambiente virtual e pode utilizar uns óculos estereoscópios que permitam a visualização em três dimensões das imagens a simular [12].



(a) HMD (retirado de [13])



(b) Monitor 3D (retirado de [14])

Figura 2.5: Exemplos de Dispositivos Visuais

Dispositivos Auditivos

A utilização de som tridimensional proporciona uma sensação de imersão bastante próxima da realidade, uma vez que o ser humano possui naturalmente audição estereofónica.

Estes sistemas permitem que o som da simulação seja recebido pelo utilizador de várias direções em seu redor, tal como se estivesse num ambiente real.

Dispositivos Físicos

Estes dispositivos permitem a simulação de outros sentidos, como o tato, a perceção de temperatura ou a tensão muscular.

No caso de sistemas táteis, é necessária a existência de *feedback*, pois estes permitem a transmissão de sensações que atuam sobre a pele, como o toque, a distinção de rugosidade, entre outras.

Além disso, existem os sistemas que atuam com forças, que dão ao utilizador a sensação de peso e pressão, utilizando braços mecânicos encaixados no corpo e que simulam o peso de objetos como se eles existissem no mundo real [12].

2.2.1.2 Dispositivos de Entrada de Dados

São os dispositivos que permitem o movimento e interação do utilizador com o mundo virtual, dando-lhe uma experiência de realidade virtual ativa, e não passiva.

Dispositivos de Interação

Existem várias formas de interagir com a simulação, sendo que as mais comuns utilizam luvas de dados (sistema mecânico capaz de reconhecer os movimentos das mãos - figura 2.6) e sensores de entrada biológicos, que processam sinais elétricos musculares através de eletrodos colocados na pele ou que usam comandos de voz.



Figura 2.6: Luva de Dados (retirado de [15])

Nos sistemas mais recentes a interação entre o utilizador e o modelo simulado é realizado através da deteção das mãos com dispositivos capazes de acompanhar o seu movimento utilizando infravermelhos, como é o caso do dispositivo *Leap Motion* (ver capítulo 3.2.2).

Dispositivos de Trajetória

Através da diferença de posição em relação a um ponto de referência, estes sistemas realizam *tracking* de partes do corpo do utilizador, analisando o seu movimento. Este pode ser realizado de forma ativa, utilizando sensores colocados sobre essas partes do corpo ou do objeto, ou passivo no caso do uso de sensores óticos ou câmaras que observam o objeto e determinam a sua orientação e posição [12].

Capítulo 3

Ferramentas Utilizadas

De forma a completar todas as tarefas referidas anteriormente, será necessário um conjunto de várias ferramentas, algumas delas físicas (*hardware*) e as restantes no domínio do *software*.

No primeiro caso, as ferramentas que serão usadas foram escolhidas tendo em conta o material existente no local de trabalho, depois de analisadas as suas características, e após ter-se confirmado que todas as funcionalidades do mesmo permitiam o desenvolvimento da aplicação pretendida.

Em relação ao *software*, este foi selecionado de acordo com as compatibilidades com os modelos biomecânicos já existentes, bem como com o formato necessário para que seja possível a implementação dos mesmos em ambiente virtual.

De seguida, são apresentadas as características gerais das ferramentas que foram usadas para o desenvolvimento da presente dissertação.

3.1 *Software*

3.1.1 Abaqus

O Abaqus é um *software* utilizado em engenharia, especializado na análise e predição de forças e deformações em estruturas, através do método de elementos finitos.

É capaz de resolver problemas lineares ou não lineares, mostrando a evolução do resultado da ação de forças aplicadas a objetos que possuam qualquer tipo de geometria ou que sejam constituídos por qualquer material, numa sequência de *frames* (ver exemplo na figura 3.1).

Além disso, possui ainda capacidade de simular problemas relacionados com transferência de calor, aquecimento de materiais condutores de eletricidade ou acústica.

É uma ferramenta simples e intuitiva para o utilizador, fornecendo acesso a múltiplos tipos de manuais, dependendo da simulação pretendida, sendo que apenas é necessário fornecer ao programa alguns dados específicos para cada caso (como por exemplo as condições limite, a geometria do modelo), e o *software* encarrega-se de escolher automaticamente a escala temporal e os ajustes de continuidade entre *frames*, de forma bastante eficiente [16].

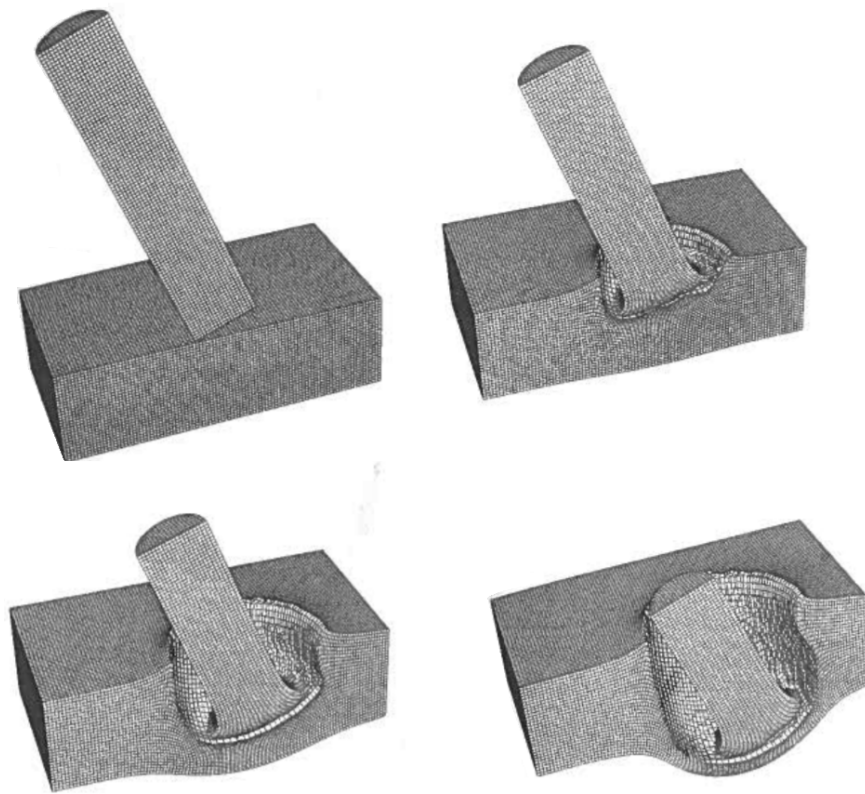


Figura 3.1: Exemplo de simulação de deformação de um objeto usando Abaqus (adaptado de [17])

3.1.2 Ficheiros STL (Estereolitografia)

Os ficheiros STL (nome proveniente de *STereoLithography*) são um dos métodos mais utilizados na representação de objetos e modelos CAD (*Computer Aided Design*) em termos da sua geometria, e descrevem de forma universal e uniformizada os objetos a representar em sistemas de prototipagem e desenho assistido por computador.

Este tipo de formato consiste numa lista desordenada de faces triangulares que pretendem aproximar a superfície, externa e fechada, de um objeto [18, 19]. Cada ficheiro STL pode ter dois formatos distintos: em ASCII (*American Standard Code for Information Interchange*) ou binário. O primeiro tem um tamanho consideravelmente maior que o segundo, mas tem a vantagem de permitir a leitura direta da informação por parte de um utilizador [18].

Cada triângulo é descrito através de três vértices e um vetor unitário normal que indicam qual é o lado da face que é visível, organizados segundo a "regra da mão direita", em que o polegar aponta na direção da normal à face.

Este tipo de ficheiros tem diversas vantagens, como o facto de ser universal e poder ser usado, de forma simplificada, por vários sistemas de prototipagem rápida ou em sistemas CAD. No entanto, possui também desvantagens, como o facto de possuir um tamanho bastante maior que o

ficheiro de dados CAD original, devido às redundâncias existentes nas faces que partilham vértices de triângulos, de modo a formar uma superfície, conforme demonstrado na figura 3.2.

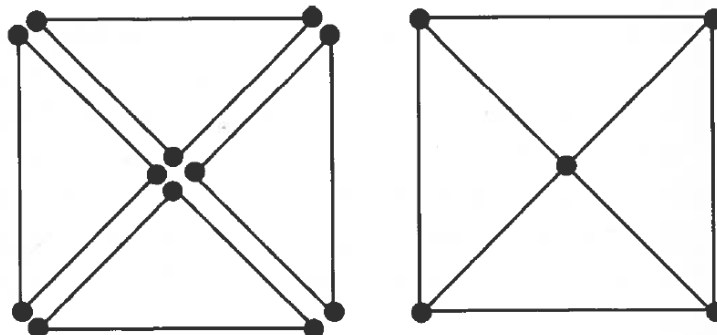


Figura 3.2: Redundância de vértices em STL. Na imagem da esquerda encontra-se representada a forma como uma superfície se encontra descrita num ficheiro STL, havendo vértices duplicados ao longo do conjunto de dados. Na imagem da direita encontra-se demonstrado como uma superfície deveria estar descrita para evitar redundâncias. (retirado de [18])

3.1.3 Virtual Reality Modeling Language

A *Virtual Reality Modeling Language*, ou VRML (pronuncia-se "vermal"), é uma linguagem de programação que permite a criação de um “espaço” dentro da rede que é a internet, de modo a simplificar o modo como o ser humano percebe a informação, fazendo-o sentir (utilizar os sentidos) em vez de ler e processar informação escrita [20].

O seu modo de funcionamento é bastante semelhante ao utilizado na criação de páginas Web através de HTML (*HyperText Markup Language*), que envia um pedido pela internet, e aguarda pelos documentos requisitados), mas recebendo como resposta um "mundo", ou seja, uma descrição de objetos e onde se encontram no ambiente.

Um ficheiro do tipo VRML contém texto, que é lido por aplicações próprias para esta linguagem (chamadas de *VRML browsers*), e que procedem à criação do ambiente descrito, permitindo a navegação dentro dele. Estes ficheiros, que contêm a extensão ".wrl"(ou *dot world*, e daí a denominação de "mundos") possuem uma constituição própria, que segue a seguinte base [21]:

- VRML header (Cabeçalho) - indica o tipo de ficheiro que é (neste caso, VRML), qual a versão que é usada no ficheiro em causa, qual o tipo de caracteres utilizado, entre outros parâmetros;
- Nós - descrevem as formas e as suas propriedades no mundo (a sua indicação é obrigatória, seguido de chavetas, possuindo *fields* dentro delas);
- Fields (opcional) - definem os atributos dos nós, como por exemplo a cor do objeto, o tamanho ou a posição;
- Comentários (opcional) - fornecem informações extra que não afetam a aparência do mundo, dando notas ao ficheiro (ex: autor, data da criação do mundo).

É possível fazer ligações entre vários mundos virtuais, utilizando a *World Wide Web* (também conhecida como WWW), assim como ligações com páginas Web, da mesma forma que num *site* comum é possível ligar uma outra página diferente. Por exemplo, usando VRML, pode-se ligar uma porta a um outro mundo criado noutra local da rede, fazendo com que o utilizador sinta que um mundo é a continuação do anterior [21].

O funcionamento desta linguagem usa princípios básicos de computação gráfica, através de duas peças básicas, chamadas de *hubs* e *spokes*. O primeiro termo refere-se figurativamente ao “centro” de uma roda de bicicleta, ao ponto onde estão ligados os aros metálicos da roda. O segundo refere-se a esses mesmo aros, ou raios, que partem do *hub* em todas as direções [22]. Juntando as duas peças referidas, são criadas as *frames* que definem a forma de um objeto a desenhar no mundo gerado pelo computador.

Ao criar o mundo, o computador lê uma lista de pontos do ficheiro de texto VRML (pontos estes equivalentes aos *hubs*), transformando-os em objetos visíveis, e liga-os uns aos outros, criando desta forma as *wireframes*. Esse processo gera as superfícies dos objetos, dando-lhes uma aparência sólida.

3.1.4 Blender

O Blender é um *software* controlado por uma instituição intitulada Blender Foundation, e trata-se de um *freeware*, *open source* [23]. Este *software* é uma ferramenta bastante poderosa e adaptável que é utilizada na criação de animações executadas com o auxílio de computador.

Disponibiliza um conjunto de instrumentos úteis em diversos casos, como por exemplo na modelação de objetos em computador, animação de modelos, simulação, renderização, entre outros [24, 25]. Além disso, possui um vasto conjunto de formatos suportados, tanto para importação como para exportação, conforme pode ser visível na figura 3.3.

A interface do Blender (ver exemplo na figura 3.4), por utilizar OpenGL (*Open Graphics Library*) na sua base, é consistente e executável em todos os principais sistemas operativos (Windows 32-bit ou 64-bit, Linux e Mac OS) [25]. Através de *scripts* em Python, o Blender pode ainda ser configurável, executar funções mais apropriadas para cada caso pretendido pelo utilizador, aumentando as suas capacidades [25, 26].

Este *software* permite que o resultado de uma animação possua como característica a alta qualidade, tendo como *input* modelos tridimensionais ou outros tipos de dados [27]. O foco da sua utilização é principalmente nas áreas da indústria dos videojogos e no entretenimento, mas também pode gerar visualizações científicas de alta qualidade [27].

O Blender é um programa que inicialmente não é muito intuitivo de utilizar nem possui um manual ou tutorial, uma vez que se encontra em constante desenvolvimento através da ajuda conseguida com as grandes comunidades e fóruns onde vários *developers* se juntam e contribuem com ideias e novas funcionalidades [25, 27].

Formato	Tipo	Extensão	Utilização
Blend	nativo	.blend	Importar/Exportar
OBJ	ASCII	.obj	Importar/Exportar
CSV	ASCII	.csv	Importar ^a
JPEG	imagem	.jpg, .jpeg	Importar ^b /Exportar
TIF	imagem	.tif, .tiff	Importar ^b /Exportar
PNG	imagem	.png	Importar ^b /Exportar
GIF	imagem	.gif	Importar ^b /Exportar
EXR	imagem	.exr	Importar
FITS	imagem/estrutura dados	fit, .fits	Importar ^a
AVI	video	.avi	Exportar
MPEG	video	.mpg, .mpeg	Exportar
MOV	video	.mov	Exportar ^c
H.264	video	.mov	Exportar ^c

^a Importado via ficheiro Python standard

^b Usado para texturas de malhas de superfície

^c Atingido em SO baseados em Linux via FFmpeg (<http://www.ffmpeg.org>)

Figura 3.3: Tipos de formatos suportados pelo Blender (adaptado de [25])

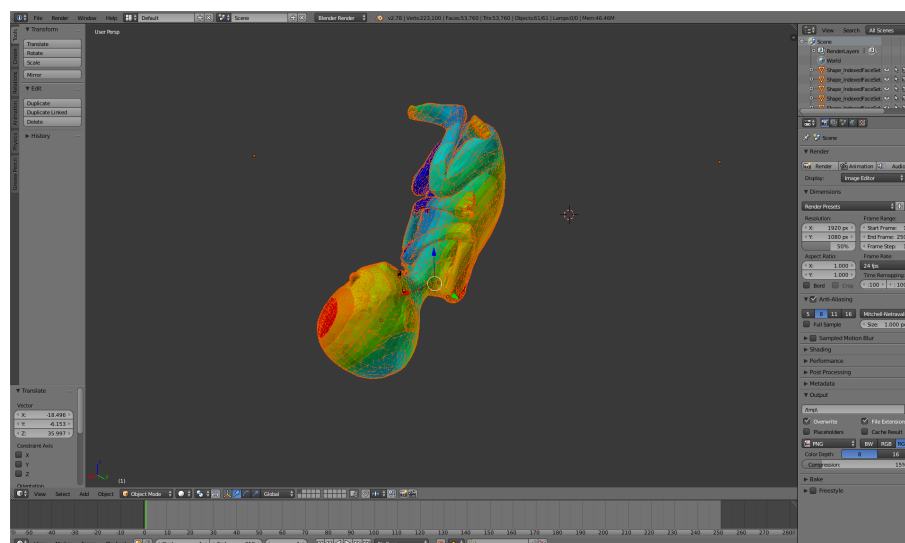


Figura 3.4: Interface do Blender

3.1.5 Ficheiros FBX

O formato FBX (ou FilmBox) é um tipo de ficheiros suportado por praticamente todos os *softwares* que utilizam motores tridimensionais. Foi criado pela empresa Autodesk, e permite a representação de modelos 3D, uma vez que suporta a descrição dos objetos através de vértices e normais. Além disso, suporta ainda animações e outros tipos de modelos [28].

A sua estrutura consiste em nós e subnós, organizados segundo o esquema em árvore, como se encontra representado na figura 3.5.

Cada um destes nós e subnós tem propriedades específicas, que o descrevem e permitem o armazenamento da informação relativa a cada triângulo que compõe a superfície de um determinado objeto. Os vértices desses triângulos possuem associados à sua descrição três coordenadas espaciais, o que indica a sua posição [28]. Já a direção em que a superfície é visível encontra-se determinada através da normal à face do triângulo, da mesma forma que acontece nos ficheiros STL.

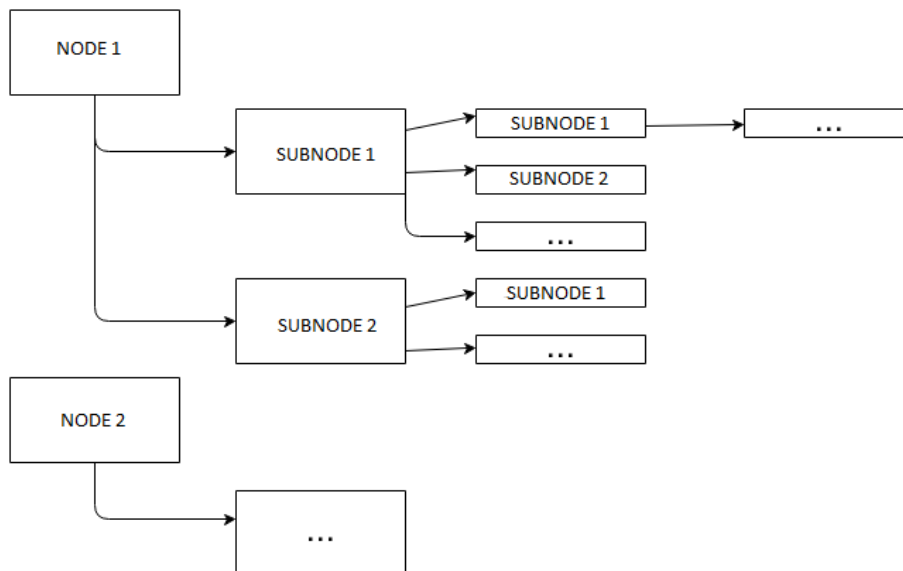


Figura 3.5: Estrutura (em árvore) de um ficheiro FBX (retirado de [28])

3.1.6 Unity

O Unity (figura 3.6) consiste numa plataforma cruzada (ou seja, *software* que recorre a vários ambientes e linguagens na sua implementação, mas que permite ao utilizador a escrita de apenas um programa) que utiliza motores de jogos para a criação de ambientes virtuais [29].

Desenvolvido pela empresa Unity Technologies, é utilizado para o desenvolvimento de videojogos para diversas plataformas (como computador, consolas ou telemóveis) [29], bem como para a construção e design de ambientes virtuais aplicáveis em áreas mais científicas [30], como a simulação de situações da área da medicina ou de engenharia. O seu uso é bastante variado, e pode ir desde a simulação virtual simples até ao desenvolvimento de sistemas complexos de, por exemplo, informação geográfica [31].

É uma ferramenta bastante robusta, uma vez que inclui um motor de física fiável no que diz respeito a elementos como terrenos, texturas, partículas, som, entre outros [30]. Além desse aspeto, possui grandes capacidades de compatibilidade com outros *softwares* e várias linguagens, bem como uma enorme variedade de publicação das cenas criadas [31]. A exportação das cenas pode ser realizada em diversos formatos, entre os quais o tipo executável (.exe).

Os ambientes criados com o auxílio do Unity permitem o suporte de periféricos para a interação, além do rato ou teclado [31], como por exemplo os óculos de realidade virtual, Oculus

Rift, ou dispositivos para simular as mãos do utilizador, com o Leap Motion, ambos descritos no subcapítulo 3.2.

De modo a melhor adaptar o ambiente virtual e os seus objetos ao caso a simular, o Unity suporta ainda o uso de *scripts* em inúmeras linguagens de programação, ou seja, códigos de comando associados a objetos do ambiente virtual e que são usados para definir o comportamento dos mesmos. Com esta associação de *scripts*, é possível definir a interatividade entre os vários objetos ou entre o utilizador e os objetos [31].

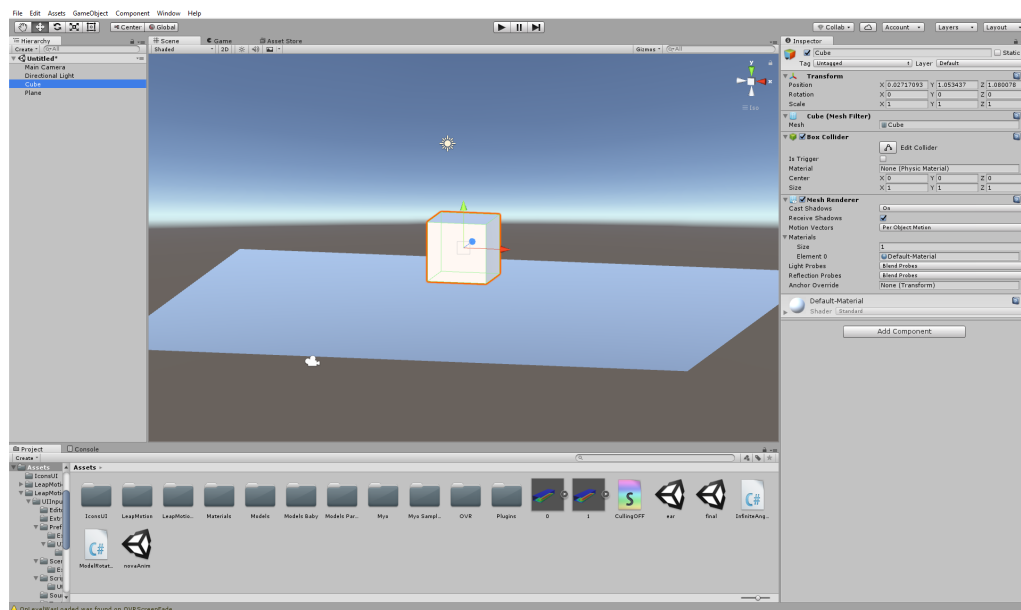


Figura 3.6: Interface do Unity

3.2 Hardware

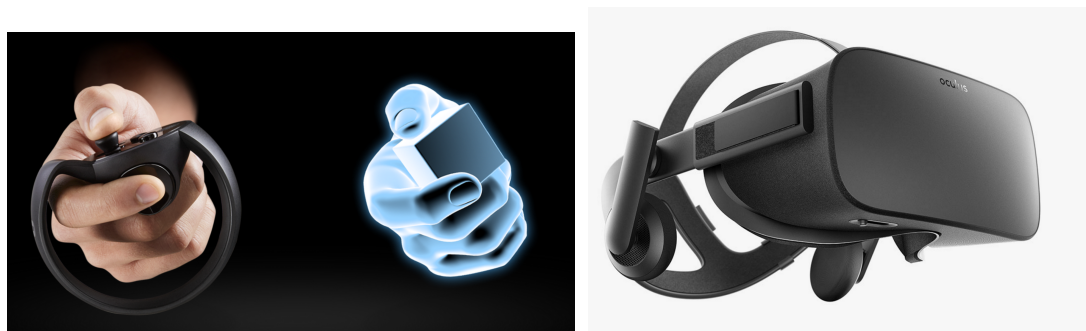
3.2.1 Oculus Rift

O aparelho Oculus Rift consiste num sistema especializado em realidade virtual capaz de isolar o utilizador do mundo real e dar-lhe uma experiência quase inteiramente imersiva (o sentido do olfato, por exemplo, não se encontra incluído) num ambiente gerado por computador, seja ele um jogo, um filme ou uma simulação científica. Por esse motivo, trata-se de um dispositivo visual do tipo HMD (ver subcapítulo 2.2.1.1).

Sendo especializados neste tipo de tecnologia, proporcionam uma sensação de presença bastante avançada, combinando ecrãs de resolução elevada com sistemas de posição e orientação da cabeça precisos. Isso auxilia o utilizador, na medida em que torna toda a simulação muito mais intuitiva.

Além das capacidades visuais que possui, pode ainda incluir dispositivos de entrada de dados, como os controladores Touch (figura 3.7a), que permitem a utilização das mãos do utilizador para controlar variáveis da simulação, aumentando assim a interatividade.

Os seus *displays* utilizam díodos emissores de luz orgânicos (OLED), constituídos por um filme orgânico cuja camada de emissão eletroluminescente emite luz de acordo com uma corrente elétrica. Essa característica torna possível a diminuição do enjoo provocado por vários dispositivos, uma vez que a imagem se torna muito mais focada neste tipo de *display*.



(a) Controladores Touch para Oculus Rift (adaptado de [32]) (b) Oculus Rift - Development Kit 2 (DK 2) (adaptado de [32])

Figura 3.7: Hardware - Oculus Rift e Controladores Touch

Para o uso do Oculus Rift aconselha-se a existência de um computador com um processador capaz de suportar as funcionalidades totais do dispositivo, bem como o uso de uma placa gráfica que consiga acompanhar a poderosa potência computacional que é necessária para que a simulação seja criada e corra de uma forma mais aproximada da realidade.

Este produto possui várias versões, das quais a mais recente e utilizada na presente dissertação, é a versão Development Kit 2 (DK 2) (ver figura 3.7b). Em comparação com a versão anterior, difere principalmente em termos de resolução do ecrã, do uso de OLED, tempo de latência (diferença entre a altura em que o *input* é introduzido no sistema e a altura em que o seu efeito é notado), frequência de atualização do ecrã, entre outras características mais físicas. Na tabela 3.1 são comparadas as características das duas versões do dispositivo.

Tabela 3.1: Comparação das Características dos DK 1 e 2 dos Oculus Rift (adaptado de [33])

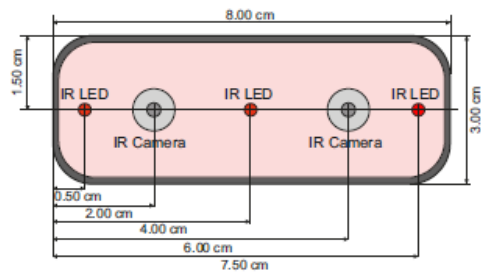
Características	DK 1	DK 2
Resolução do ecrã	1280x800	1920x1080
Disposição de Pixels	RGB	Pentile
OLED	Não	Sim
Tamanho do ecrã	7"	5.7"
Latência	50-60ms	20-40ms
Frequência de Atualização	60Hz	75Hz
Deteção de Orientação	Sim	Sim
Deteção da Posição	Não	Sim

3.2.2 Leap Motion

Este dispositivo (figura 3.8a), criado pela empresa com o mesmo nome (Leap Motion) [34, 35], permite a manipulação de objetos no monitor do computador ou o controlo dos mesmos através de gestos e movimentos executados com as mãos do utilizador. Trata-se de um aparelho recente (lançado em 2013 [36]), com as dimensões 7,6 cm x 3 cm x 1,3 cm, e constituído internamente por duas câmaras detetoras de infravermelhos e três LEDs infravermelhos (ver figura 3.8b) [36].



(a) Dispositivo Leap Motion



(b) Visão esquemática do Leap Motion (retirado de [35])

Figura 3.8: Leap Motion

Este aparelho foi construído com o objetivo de incluir uma nova funcionalidade bastante conveniente nos sistemas de realidade virtual: aumentar a intuitividade e imersividade do utilizador no manuseamento dos ambientes virtuais [37].

Para executar a função da deteção das mãos e dedos do utilizador, lança feixes de luz infravermelha no objeto (neste caso, as mãos), criando um espaço tridimensional em frente do dispositivo, capaz de acompanhar a presença e movimento de tudo o que se atravessa nele, convertendo-o em dados, nomeadamente posição e orientação da mão e dedos [36, 37]. No entanto, o modo exato de deteção não é conhecido, sendo que apenas são conhecidos os componentes referidos.

O Leap Motion faz parte de uma categoria de dispositivos de rastreamento ótico, que ao contrário dos sistemas mecânicos (como as luvas de dados), é uma opção bastante mais barata, mais eficiente e não invasiva [37]. O movimento captado é discretizado numa sequência de estados, com movimentos incrementais entre cada um, formando *frames*. Estes *frames* são depois analisados no computador, criando-se um movimento contínuo [37].

Apesar de atualmente ainda possuir várias limitações, como a elevada sensibilidade às condições luminosas, situação melhorada com os recentes sensores mais robustos, continua a ser um dos dispositivos com melhor precisão, quando comparada com equipamentos semelhantes (possui em média uma precisão de 0,7 mm, enquanto que o dispositivo Microsoft Kinect Controller possui em média uma precisão de 1,5 cm) [37, 35]. Uma das principais limitações do Leap Motion, ainda a ser combatida, é a oclusão de segmentos da mão. Este problema ocorre quando alguns segmentos da mão ficam ocultos por outros, e os sensores do Leap Motion não os conseguem detetar. Uma forma para prevenir esta questão é a total abertura das mãos por parte do utilizador, ou a orientação das mesmas estar virada diretamente para o dispositivo [37].

De modo a colmatar esta limitação, existem já algumas funções implementadas no SDK (*Software Development Kit*) existentes no site de desenvolvimento do Leap Motion (ver [38]), que estimam a posição dos segmentos com base em posições anteriores e comparando essas estimativas com os segmentos detetados.

Para cumprir as suas funções, o Leap Motion disponibiliza um site com a API (*Application Programmer Interface*) em várias linguagens de programação (C++, C# - para Unity, Java, Python, entre outras), e que permitem o rastreamento das pontas dos dedos, da palma da mão e do esqueleto (falanges e articulações) [36, 37]. Além disso, é possível ainda a deteção de alguns gestos específicos, e associá-los a funções. As funcionalidades existentes na API podem ser observadas na tabela 3.2.

Tabela 3.2: Funções disponibilizadas na API do Leap Motion [37]

Categoria	Função(ões)
Mão	Deteção da posição da palma, normal, direção e dedos associados
Dedo	Deteção da posição e orientação de cada osso do dedo
Gestos	Deteção do gesto de círculo, toque (para a frente e para baixo com o dedo) e deslize
Outras	Distinção entre mão esquerda e direita
	Obtenção de informação através dos nomes dos dedos

Na figura 3.9 podem ser observados os quatro tipos de movimentos que são atualmente reconhecidos pelas funções fornecidas na API do Leap Motion. Através de programação, podem ser definidos outros gestos, cenário que depende do utilizador e do ambiente virtual que ele cria.

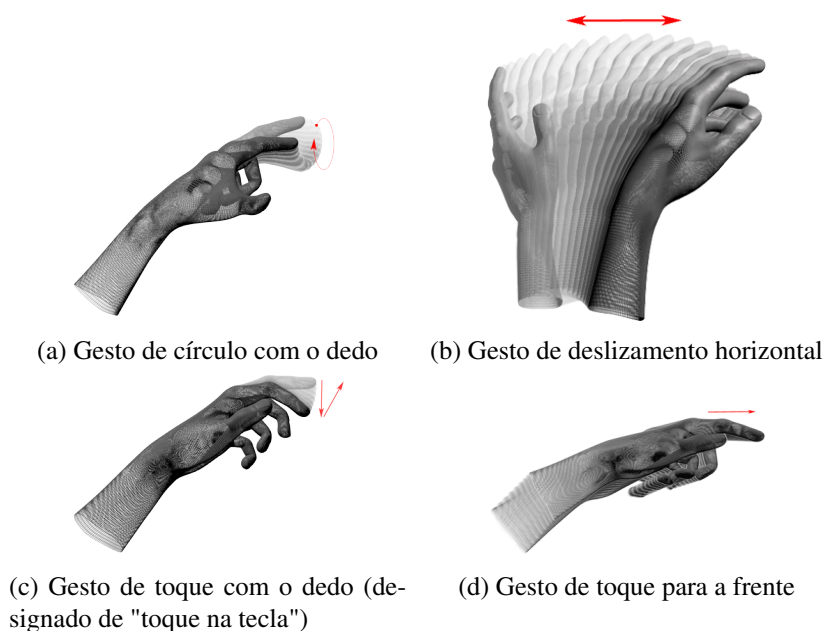


Figura 3.9: Gestos reconhecidos pelas funções fornecidas na API do Leap Motion (retirado de [39])

Capítulo 4

Biomecânica do Parto

A biomecânica é uma área que investiga o movimento de um ponto de vista mecânico e físico (avaliando a cinemática e dinâmica), aplicando esse estudo ao corpo humano, no que diz respeito aos tecidos moles ou às restantes estruturas anatómicas, bem como dos efeitos que as forças aplicadas têm nos sistemas biológicos.

No caso retratado na presente dissertação, será abordado o parto humano, através do canal vaginal, sendo que o foco principal do estudo irá incidir nas consequências que ocorrem a nível muscular no pavimento pélvico.

Nos subcapítulos seguintes serão descritos quais os objetivos, metodologias e resultados obtidos em análises biomecânicas já realizadas no caso de estudo referido, bem como a evolução que ocorreu ao longo do tempo na investigação do parto, e ainda a justificação da escolha do modelo que foi utilizado na presente dissertação.

4.1 Objetivos do Estudo Biomecânico do Parto

Como será descrito nos subcapítulos 4.3 e 4.4, o parto vaginal no ser humano é um fator de risco significativo no que diz respeito ao aparecimento de lesões e disfunções do pavimento pélvico [40]. Assim, torna-se necessário distinguir alguns tópicos importantes para que estas lesões e disfunções sejam minimizadas.

Ao longo dos últimos anos foram realizados alguns estudos, cujos objetivos visaram o entendimento das consequências que ocorrem nos músculos do pavimento pélvico (doravante designados por MPP), bem como possíveis resoluções ou prevenções para as mesmas.

Desta forma, os primeiros estudos do parto do ser humano foram focados na quantificação das forças que atuam nos MPP, e posteriormente estenderam-se também ao feto. Após a sua determinação, os estudos convergiram na procura da diminuição das consequências das forças opostas ao movimento de saída do feto durante um parto, de forma a facilitar esse mesmo movimento, minimizando assim as eventuais complicações descritas ao longo do presente capítulo [41].

Além disso, o estudo biomecânico do parto humano permite que novas técnicas sejam desenvolvidas, de modo a reduzir de forma mais eficaz as deformações causadoras de lesões nos MPP, bem como o desenvolvimento e aumento das potencialidades dos modelos computacionais e numéricos existentes [42, 43]. Estes modelos permitem que os profissionais da área da saúde possam tomar decisões clínicas antes de um determinado procedimento ser aplicado na realidade, o que permite um acompanhamento mais personalizado a cada caso e uma prevenção maior no que diz respeito aos traumas nos MPP [40].

Para concluir, os estudos realizados pretendem melhorar a saúde e estilo de vida da mulher, assim como reduzir os custos médicos dos procedimentos utilizados atualmente. Em relação às especialidades existentes, pretende-se ainda que as práticas clínicas sejam melhoradas, em particular quando relativas à obstetrícia [44, 45, 46].

4.2 Evolução do Estudo do Parto

Desde que a tecnologia começou a ser desenvolvida que o ser humano a aplica no estudo do parto, de forma a reduzir as comuns complicações que aconteciam durante o processo e de modo a que a prevenção das disfunções seja melhorada. Os métodos de estudo do parto foram assim bastante diversos ao longo do tempo, mas permitiram o desenvolvimento do estudo da anatomia e das funções das estruturas pélvicas. Essa evolução será descrita ao longo do presente subcapítulo.

Após a percepção de que um dos principais fatores de risco durante um parto era a dimensão do crânio do feto e a excessiva dilatação muscular da zona pélvica provocada pela passagem do mesmo, os primeiros estudos concentraram-se nas chamadas proporções céfalo-pélvicas, ou seja, na simples medição do diâmetro e perímetro do crânio do feto, e na comparação desta medição com o tamanho da pelve materna [40].

Conforme seria de esperar, este método não é suficiente para se poder prever as possíveis lesões que naturalmente ocorrem durante um parto, pelo que se sentiu a necessidade de evoluir a tecnologia utilizada pela medicina neste estudo. Desta forma, o processo foi posteriormente estudado através da aplicação de técnicas de imagem, nomeadamente a utilização de ultrassons e ressonância magnética [47, 48].

A aplicação destas técnicas permitiu a transformação das imagens obtidas em modelos computacionais para posterior análise mecânica, o que ajudou na quantificação da biomecânica do parto. Através da aplicação de contraste durante uma ressonância magnética, é possível a observação dos tecidos moles utilizados durante um parto. No entanto a resolução espacial é relativamente baixa e não permite a identificação das estruturas do pavimento pélvico com o detalhe necessário para o estudo exato do processo do parto, como por exemplo a identificação das lesões nas fibras musculares ou dos tecidos conjuntivos [49].

De modo a melhor compreender o comportamento que os tecidos teriam durante o processo do nascimento do feto, o passo seguinte consistiu no estudo em cadáveres e da análise das fibras musculares. Isso permitiu o desenvolvimento das simulações em computador que começaram a

surgir através da aplicação de modelos matemáticos que procuravam descrever o modo de funcionamento das estruturas com interesse para o caso de estudo.

Com o aperfeiçoamento dos modelos utilizados, a facilidade de criação de novas simulações com tecidos moles virtuais próximos dos reais aumentou, e incluiu novos métodos, como por exemplo o método dos elementos finitos na modelação dos músculos [50, 51, 52]. Esta evolução originou o crescimento do interesse da biomecânica na área dos tecidos moles, crescimento esse que se mantém ainda nos dias de hoje.

Uma vez que a tecnologia atualmente se encontra bastante mais desenvolvida, os modelos tridimensionais que começam agora a surgir são uma promessa na representação do comportamento mecânico dos MPP durante o parto [53, 54, 55, 56], o que permite a previsão das lesões e disfunções, dando lugar a novas formas de as evitar.

No entanto, estes modelos não estão ainda na posição desejada, que permita a identificação de todas as perturbações que ocorrem nas estruturas da pelve feminina ou no próprio feto, e por isso torna-se necessário listar quais os pontos a serem melhorados para que os modelos possam evoluir. Desta forma, os principais desafios encontram-se na modelação da geometria complexa dos MPP e dos mecanismos da cabeça do feto. Esta modelação complica-se também devido à escassez de dados experimentais relativos às estruturas do pavimento pélvico e à validação clínica dos modelos biomecânicos utilizados [42, 43].

Desta forma, a avaliação dos danos continua apenas a poder ser executada através de técnicas como a palpação, eletromiografia ou ressonância magnética (o principal parâmetro a avaliar é a capacidade de contração muscular, uma vez que a contratilidade dos músculos pode estar associada a disfunções), sendo os modelos apenas utilizados para a previsão dos locais mais prováveis onde acontecerão as lesões.

4.3 Análise Biomecânica do Parto

Conforme foi referido no subcapítulo 4.2, os modelos e técnicas utilizados para o estudo biomecânico do parto não se encontram ainda totalmente desenvolvidos, particularmente no que diz respeito ao conhecimento completo das complexas funcionalidades da zona pélvica, apesar de todas as especialidades da medicina existentes neste campo (como a obstetrícia, ginecologia, urologia, entre outras) [45, 46].

O estudo biomecânico do parto iniciou-se com a modelação das estruturas envolvidas, nomeadamente dos MPP, de forma a poder estimar-se as alterações mecânicas provocadas pela passagem do feto, com o objetivo de se poderem avaliar as forças envolvidas no processo. Essa modelação foi realizada através do método dos elementos finitos, que permite a representação dos efeitos provocados pela passagem do feto pelo pavimento pélvico. A informação geométrica dos músculos foi obtida a partir de um cadáver de uma mulher de 72 anos [57].

Além disso, foi também incluído um corpo de um feto na simulação, tendo a cabeça do mesmo sido modelada de modo a incluir a pele, os tecidos moles e o crânio (com as respetivas suturas cranianas e as fontanelas - figura 4.1) [40, 58]. A inclusão dessas estruturas permite a deformação

da cabeça (de forma semelhante ao que acontece na realidade), tornando-a mais alongada à medida que o feto avança no canal vaginal.

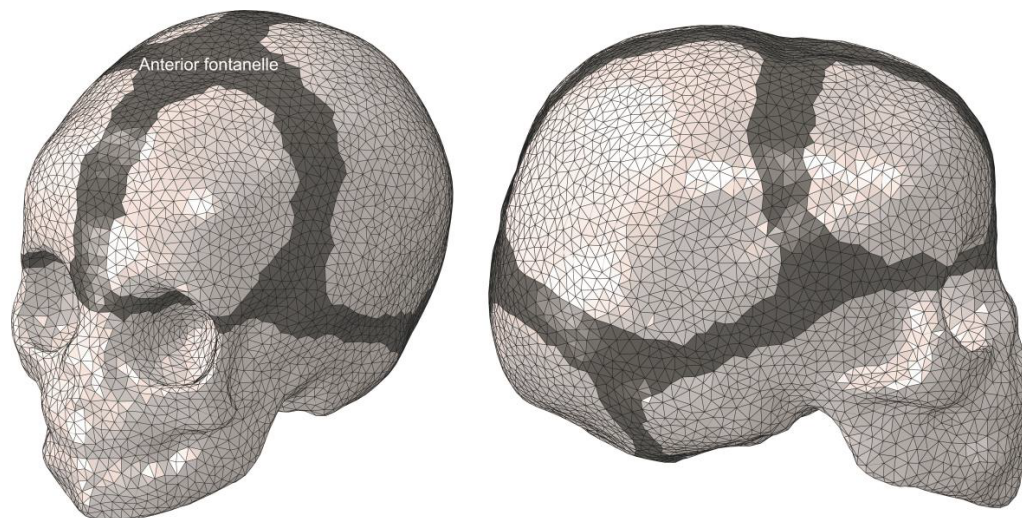


Figura 4.1: Modelo do crânio do feto. A branco encontra-se representada a superfície óssea, e a preto as regiões correspondentes às fontanelas e suturas. (retirado de [40])

A simulação foi desenvolvida de forma a não limitar o movimento do feto, permitindo que o número de graus de liberdade fosse o maior possível, de modo a que o trajeto realizado pelo feto fosse o mais natural possível, tendo apenas como entraves as obstruções anatómicas do canal de nascimento e a anatomia da pelve.

O movimento vertical descendente do feto foi controlado através da definição de pontos de referência ao longo do seu corpo (ver figura 4.2). Um deles foi atribuído a um grupo de elementos no interior do crânio do modelo, tendo este grupo sido considerado rígido [40].

Desta forma, consegue-se simular o parto com a garantia de que as únicas forças envolvidas ao longo dos vários passos fossem as criadas pela contração abdominal e uterina da parturiente, e as forças de resistência causadas pelas obstruções anatómicas da cavidade pélvica, sendo que estas segundas são dependentes das características físicas do feto e do canal de nascimento (como o tamanho do crânio do feto, a forma das diferentes estruturas ou a elasticidade dos tecidos) [59].

De modo a analisar apenas o impacto do parto nos MPP, foi assumido no estudo elaborado que o feto se apresentaria na posição mais favorável, e a simulação foi criada utilizando materiais criados e que possuem propriedades próximas da realidade, nomeadamente a nível dos tecidos conjuntivos e do pavimento pélvico [40].

A nível dos MPP, estes foram caracterizados utilizando um modelo hiperelástico, quase incompressível, transversalmente isotrópico (modelo constituído através de inúmeras equações que determinam o comportamento muscular). Este material permite a simulação do músculo, que naturalmente possui fibras, colagénio e uma matriz extra-celular sólida, e que é um material com um comportamento não-linear, como grande parte dos tecidos biológicos [50, 53].

Para simplificar a análise dos resultados, assumiu-se ainda que o osso púbico, as extremidades do músculo elevador do ânus e a fáscia pélvica (inserida no osso púbico) se encontram fixos. Além

disso, todo o modelo esquelético existente foi assumido rígido e os músculos foram modelados com uma só espessura ao longo de todo o seu volume. Algumas das estruturas de suporte que existem naturalmente no corpo humano foram também ignoradas, assim como alguns parâmetros que podem afetar o progresso do parto (mobilidade da uretra, distância entre a sínfise púbica e a bexiga, entre outros) [40].

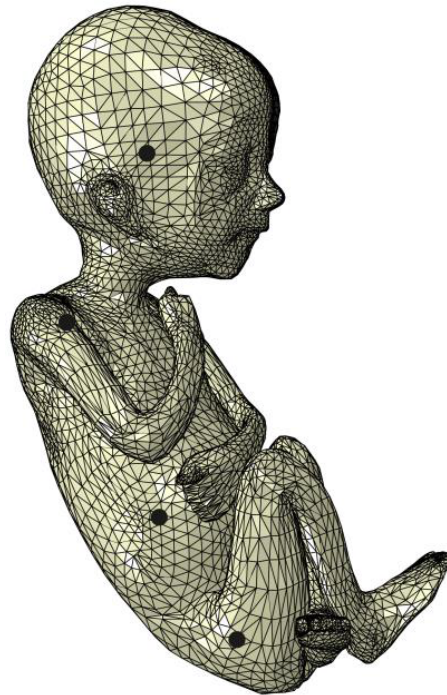


Figura 4.2: Representação dos pontos de referência do feto, para controle do seu movimento vertical (retirado de [40])

4.4 Disfunções Resultantes do Parto e Conclusões Obtidas pelos Estudos

Conforme referido no subcapítulo 4.3, o parto através do canal vaginal é um dos fatores de risco que contribuem para o aparecimento de lesões na zona pélvica, principalmente quando se trata do primeiro parto. Estas disfunções surgem mesmo quando se trata de um parto normal, sem eventos ou incidentes inesperados, provocando lesões nos MPP que afetam em média 10% dos músculos envolvidos [40].

As complicações mais comuns são o prolapso de órgãos pélvicos e a incontinência urinária ou fecal, mas existem outras, que podem estar relacionadas com o feto ou com a parturiente.

No caso do feto, as lesões são provocadas pela compressão que o corpo inteiro sofre e que podem afetar o crânio, a cara, os órgãos intra-abdominais, os nervos periféricos, a medula espinal ou o esqueleto. Estas lesões podem levar a problemas mais sérios, como fraturas ósseas (as mais

comuns incluem as clavículas ou os fêmures) ou paralisia facial [60, 61, 62, 63]. No entanto é possível concluir com as simulações que a moldagem da cabeça do feto, possível graças à existência de suturas cranianas e fontanelas, auxilia o progresso do parto, uma vez que permite a adaptação da estrutura causadora da maior deformação dos músculos à geometria do canal vaginal [64].

Relativamente à parturiente, os danos incluem as já referidas lesões traumáticas nos tecidos do pavimento pélvico (ao nível dos músculos, nervos ou tecidos conjuntivos, cuja contração e relaxamento controlados são importantes para o suporte de todas as estruturas pélvicas), e ocorrem devido ao impacto mecânico que existe durante o parto. Como consequência, a capacidade de funcionamento das estruturas do pavimento pélvico é reduzida [40]. Segundo dados clínicos, cerca de 10% a 30% das mulheres com parto natural apresentam danos severos no músculo elevador do ânus, inserção esta que é um dos casos mais difícil de reparar, e que representa portanto um sério risco de saúde [65, 66].

A evolução biológica contribuiu já com algumas formas de reduzir os danos, como por exemplo a capacidade de deformação do crânio do feto (possível graças à não consolidação total dos ossos da cabeça do feto), o que facilita o movimento ao longo do canal vaginal.

No entanto existem ainda muitas lesões que ocorrem durante o parto, pelo que é importante a procura de novos métodos que permitam o controlo da ocorrência destas lesões. Um desses métodos é designado de episiotomia, e consiste numa incisão cirúrgica no períneo que aumenta a abertura vaginal e facilita o processo do parto, reduzindo os danos a uma área menor [40]. Em [40] foram realizados estudos que determinaram o melhor ângulo e comprimento para se efetuar a incisão, ou seja, qual a melhor orientação do corte, que poderá levar ao menor número de danos ou a uma redução da gravidade das lesões posteriores ao parto (ver figura 4.3).

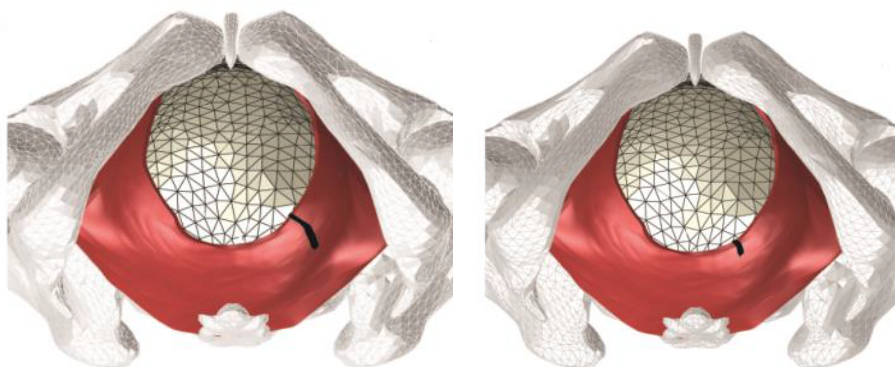


Figura 4.3: Simulação da episiotomia. À esquerda a incisão foi realizada a 60° do eixo vertical, no sentido positivo, e com comprimento de 30 mm. À direita a orientação foi de 45° e o comprimento do corte foi de 10 mm. (adaptado de [40]).

Apesar da existência deste procedimento, observam-se danos nos MPP nos partos com e sem episiotomia. A diferença é que no caso da não execução da incisão, todo o músculo elevador do ânus é danificado, enquanto que no caso com a incisão o dano é reduzido a algumas fibras (e esta conclusão já se encontra validada por simulações numéricas) [40, 67].

Porém, a realização de episiotomia está associada a uma recuperação mais lenta e menos completa da força muscular original, apesar de prevenir lacerações não controladas durante o parto, e o seu uso encontra-se ainda em debate, pois os benefícios não foram completamente comprovados.

Graças aos modelos biomecânicos, utilizando o método dos elementos finitos, é possível prever quais serão os pontos em que a probabilidade de ocorrerem lesões é maior. De acordo com os estudos realizados em [65, 66], 79% dos MPP sofrem rotura das fibras musculares durante a passagem da cabeça do feto (ou seja, no momento em que a distensão é máxima).

As simulações permitiram também identificar que os pontos com os níveis máximos de tensão e deformação são os pontos de inserção do músculo elevador do ânus no cóccix e na sínfise púbica, e que os danos ocorrem na zona mediana dos músculos [40]. Em relação aos órgãos da zona pélvica, concluiu-se que estes não serão os mais afetados pela força dos MPP, mas foram observados grandes deslocamentos durante o parto. No entanto, o estudo completo não foi executado até ao momento uma vez que os órgãos não foram incluídos na simulação de tensões (apenas em simulação de deslocamento, importante para o diagnóstico de incontinência por exemplo) [40].

Sabendo quais são os pontos mais prováveis de ruturas, é possível a existência de maior personalização dos procedimentos a executar a cada paciente, apesar de todas as limitações e variáveis dos modelos biomecânicos que são usados atualmente (como o uso ou não de epidural, que diminui a presença ativa dos músculos, ou o formato dos músculos, que varia de pessoa para pessoa). Além disso, a previsão dos danos em cada caso levaria a que os métodos de recuperação muscular fossem adaptados às lesões correspondentes, como por exemplo a implantação de malhas de reforço nos músculos, o que permitiria readquirir as funções e a força do músculo, minimizando as consequências [40].

Apesar da não inclusão de algumas das estruturas de suporte e de se terem considerado algumas simplificações, os resultados obtidos pelas simulações são bastante aproximados daqueles que foram obtidos clinicamente (os valores das forças resultantes nos músculos encontram-se na ordem dos 180-200 N, o suficiente para causar a rutura de fibras musculares [40, 68]), pelo que se pode concluir que o modelo biomecânico criado é uma ferramenta importante e credível para a clarificação dos mecanismos do parto. As análises foram realizadas logo após a simulação do parto, e sabe-se por observação clínica que as lesões são diferentes ao longo do tempo, pelo que o comportamento muscular durante a recuperação não foi avaliado [40].

4.5 Modelo Biomecânico Utilizado

Tendo em conta as diferentes simulações que foram realizadas nos estudos referidos, e no âmbito da presente dissertação, optou-se por criar a aplicação em realidade virtual utilizando o modelo biomecânico mais simples (contudo completo, excetuando algumas estruturas consideradas pouco relevantes), e sem a inclusão de episiotomia.

O modelo escolhido, existente no formato STL, permite que o *software* Abaqus corra uma simulação em que o utilizador pode seleccionar qual o tipo de análise pretendida (deformação, deslocamento, entre outros), incluir as diferentes estruturas anatómicas do parto humano ou seleccionar *frames* específicas da simulação, entre inúmeras outras funcionalidades.

No caso concreto da presente dissertação, optou-se por separar os MPP do feto, pelo que numa primeira fase apenas foram seleccionadas as estruturas relativas ao pavimento pélvico, tendo sido escolhida a opção de visualização do deslocamento muscular em relação ao ponto inicial (que se assume ser de repouso). Este tipo de simulação coloca cada face triangular do músculo com uma cor, correspondente ao nível de deslocamento espacial em valores absolutos, que no caso considerado varia no intervalo [0; 199,9 mm] (ver figura 4.4). Este intervalo, automaticamente gerado pelo Abaqus, não permite uma visualização muito precisa do valor de deslocamento de cada ponto do músculo, uma vez que a escala fica configurada para o deslocamento máximo de todo o modelo (que neste caso pertence ao feto).

No entanto, é possível observar na figura 4.5 as estruturas do pavimento pélvico, não considerando o deslocamento do feto, resultando numa variação do intervalo entre os valores 0 e 44,54 mm, que correspondem respetivamente ao deslocamento mínimo e máximo dos MPP. Este modelo, com os valores respetivos aos MPP, não foi incluído na aplicação desenvolvida, conforme será referido no capítulo 5. Este último valor referido encontra-se truncado por limitações do *software*, não tendo sido possível a determinação do valor máximo real de deslocamento dos MPP. Neste modelo, as estruturas seleccionadas incluem o corpo completo do feto e alguns tecidos moles da cabeça.

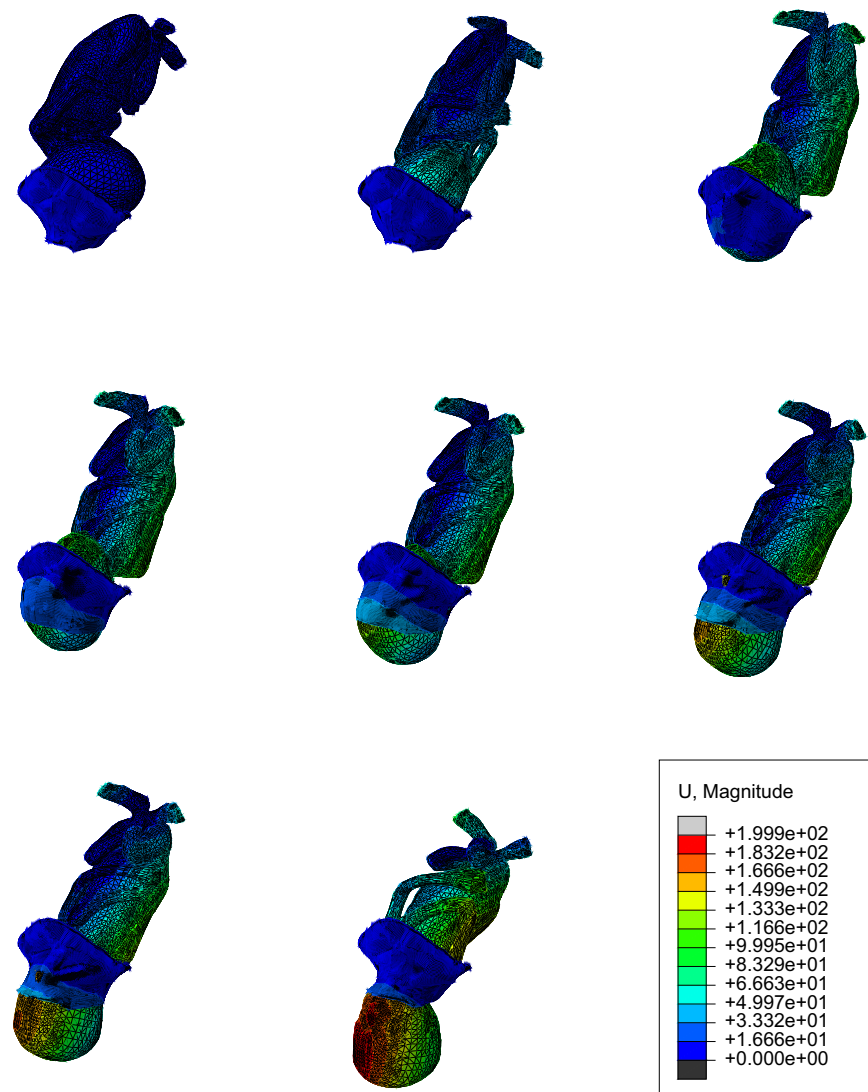


Figura 4.4: *Frames* utilizados na simulação do parto incluindo a análise do deslocamento (U) do feto (modelo utilizado na aplicação). Na última imagem é possível observar o valor dos deslocamentos mínimo (0,0 mm) e máximo (199,9 mm) do feto.

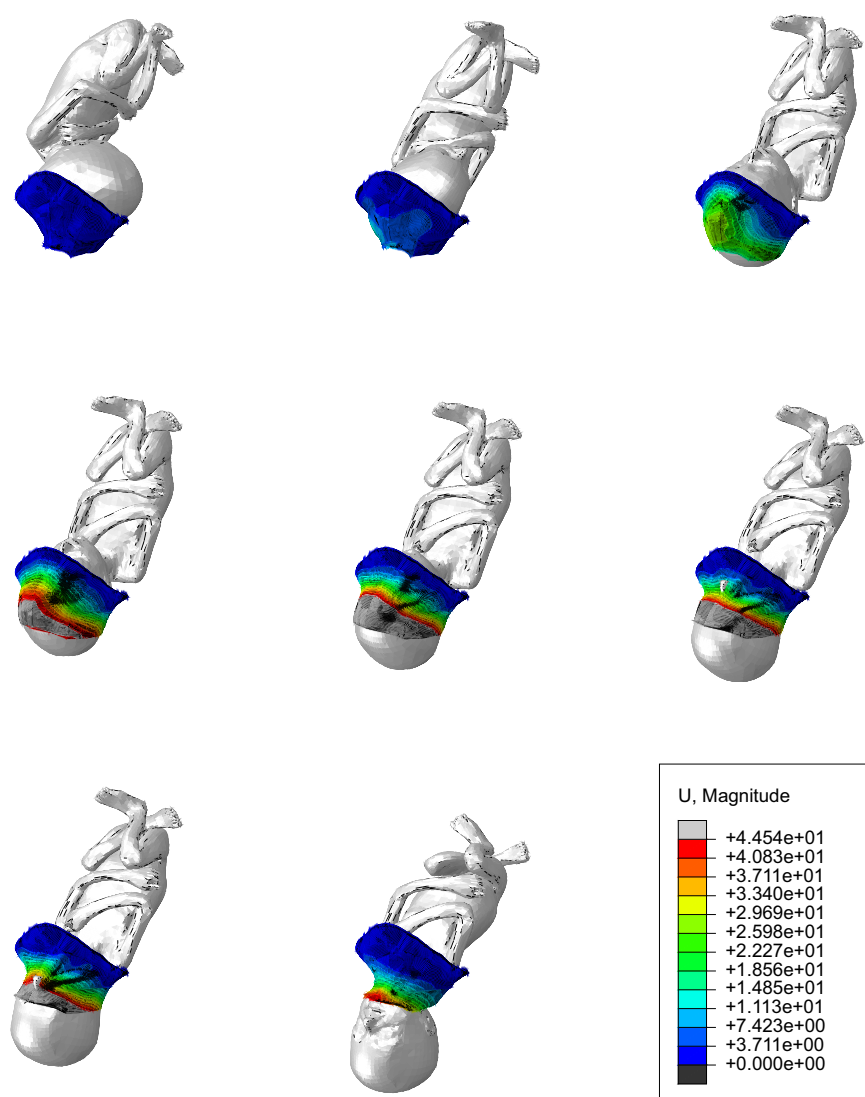


Figura 4.5: *Frames* utilizados na simulação do parto em relação ao deslocamento (U) dos MPP (feto não considerado na análise). Na última imagem é possível observar os valores do deslocamento mínimo (0,0 mm) e máximo (44,54 mm) dos MPP.

Capítulo 5

Importação do Modelo Biomecânico para a Realidade Virtual

De modo a que se possa criar um mundo virtual, é necessário que todos os objetos pertencentes ao cenário desejado estejam em formatos compatíveis com o *software* de realidade virtual utilizado. No caso da presente dissertação, o objetivo é a criação de uma aplicação que utilize realidade virtual para visualização e interação com um modelo biomecânico de um parto humano, através do canal vaginal.

Para o concretizar, é necessária a conversão do modelo biomecânico, num formato em STL, em objetos do tipo FBX, diretamente compatíveis com o software de criação da cena virtual, o Unity. Esta conversão não é direta, necessitando de uma transformação para um formato intermédio, o tipo VRML.

Ao longo deste capítulo irão ser descritos os vários passos utilizados nesta conversão, resultando num mundo virtual contendo a simulação do modelo biomecânico. O produto resultante desta conversão ainda não permite a interação com o modelo (apenas permite a visualização da simulação). O processo de criação das funcionalidades de interação encontram-se descritas no capítulo 6.

5.1 Conversão para o Formato VRML

O modelo biomecânico referido no subcapítulo 4.5 resultou, em estudos anteriores, num ficheiro do tipo OpenDocument database (ODB) e possui toda a informação obtida pelo método dos elementos finitos. Nele estão contidas as superfícies dos MPP no formato STL, capazes de ser interpretadas pelo *software* Abaqus (figura 5.1).

Após abrir o ficheiro, o utilizador pode escolher o tipo de análise que pretende realizar. No caso utilizado na criação da aplicação da presente dissertação, foi escolhida uma análise dos deslocamentos do feto e dos MPP, selecionando a variável primária U (correspondente ao deslocamento espacial em cada nó, ou seja, em cada vértice das faces triangulares que compõem as superfícies dos objetos), como se pode observar na figura 5.2.

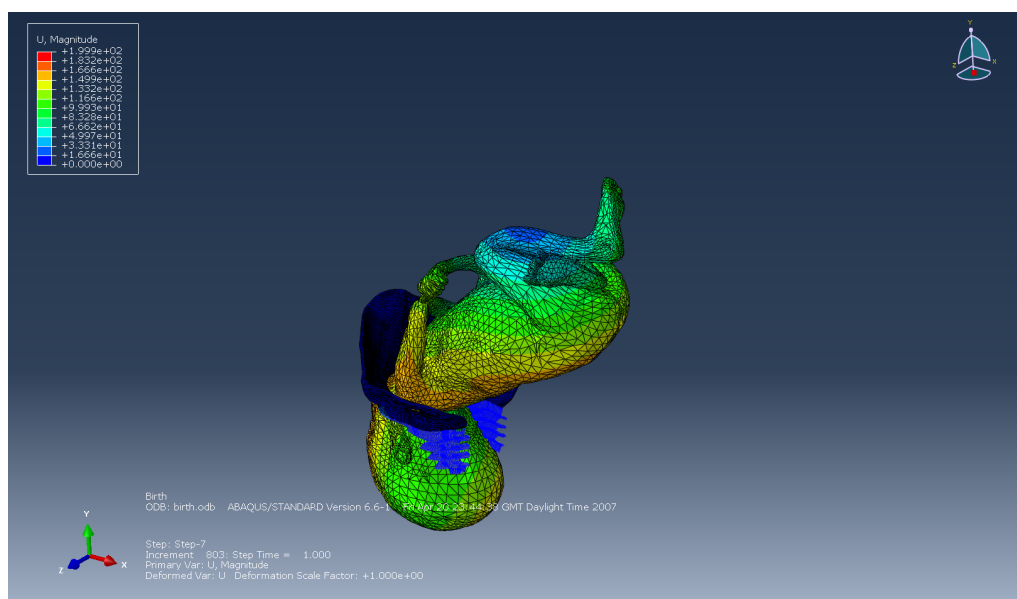


Figura 5.1: Modelo biomecânico no ambiente do *software* Abaqus

Como se pode observar na figura 5.1, existem estruturas além dos MPP no modelo, e portanto torna-se necessário selecionar os componentes a exportar para o formato VRML (um dos três formatos permitido pelo Abaqus, e o único capaz de guardar a informação relativa ao nível de deslocamento de cada um dos nós que permite colorir cada face triangular). Uma vez que é mais vantajoso para a aplicação que o feto e os MPP sejam objetos separados (para que possam ser tratados como entidades individuais), optou-se pela exportação dos componentes modelados correspondentes apenas ao pavimento pélvico. Desta forma, pode-se ver na figura 5.3a a seleção do pavimento pélvico, e na figura 5.3b o resultado desta seleção. É de notar que os componentes referentes aos tendões e tendões duros (não selecionados na figura 5.3a) se encontram no resultado, uma vez que estão incluídos num dos componentes do pavimento pélvico já selecionado.

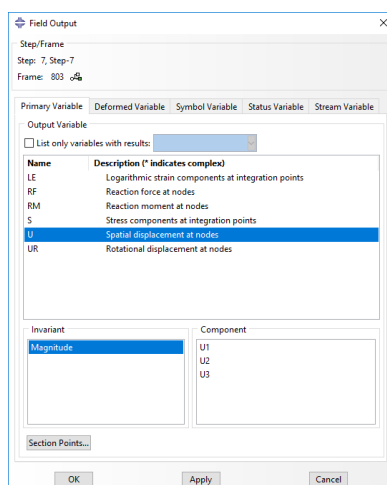
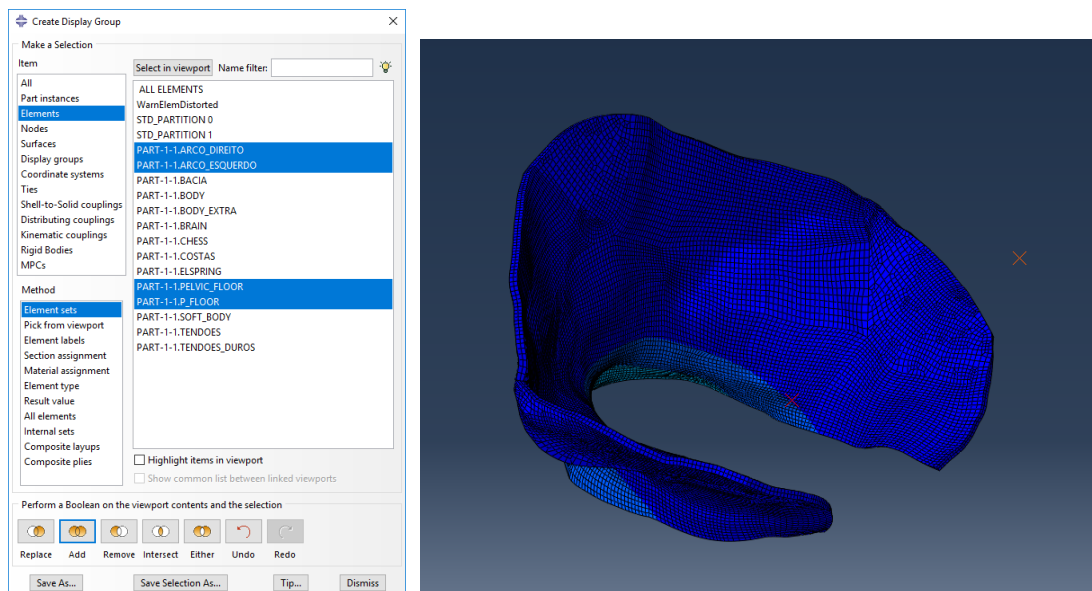


Figura 5.2: Janela de seleção da variável de análise no *software* Abaqus

Dentro do Abaqus é possível a movimentação do objeto para a posição e orientação pretendida para que o utilizador possa analisar melhor as estruturas selecionadas. No entanto, como o objetivo é a exportação para um ficheiro VRML, estas variáveis em nada influenciam o resultado.



(a) Janela de seleção dos componentes a incluir no modelo (evidenciados encontram-se os elementos correspondentes aos MPP)

(b) Resultado da seleção dos componentes - MPP simulados

Figura 5.3: Seleção dos componentes dos MPP

Após a obtenção do pavimento pélvico, é necessária a seleção das *frames* da simulação. O modelo utilizado possui 7 *steps*, que são porções do processo do parto, cada uma contendo dezenas (ou centenas) de *frames*. Cada uma destas últimas possui, para o instante de tempo correspondente, o deslocamento para cada um dos nós da superfície dos MPP em relação ao referencial (que no caso retratado se trata do *frame* zero do *step* zero).

Para a aplicação em realidade virtual, optou-se por selecionar o primeiro *frame* de cada *step* e o último *frame* do último *step*, de modo a conseguir-se uma evolução progressiva da simulação do parto. Esta seleção das *frames* deveria ser realizada automaticamente, mas devido a restrições temporais (conforme será descrito no capítulo 7.2) a seleção foi realizada manualmente.

Na figura 5.4 é possível observar os oito *frames* escolhidos e exportados para o formato VRML, correspondentes aos MPP, para posterior integração na aplicação de realidade virtual.

Concluída esta fase, resta somente a exportação dos *frames* correspondentes aos já retirados para os MPP, mas desta vez convertendo o feto para o formato VRML. Para obter este resultado, o procedimento é o mesmo: é necessário selecionar os componentes do feto pretendidos (ver figura 5.5), realizar a seleção do primeiro *frame* de cada *step* e do último *frame* do último *step* e exportar cada um deles para o formato VRML, resultando nos *frames* observados na figura 5.6.

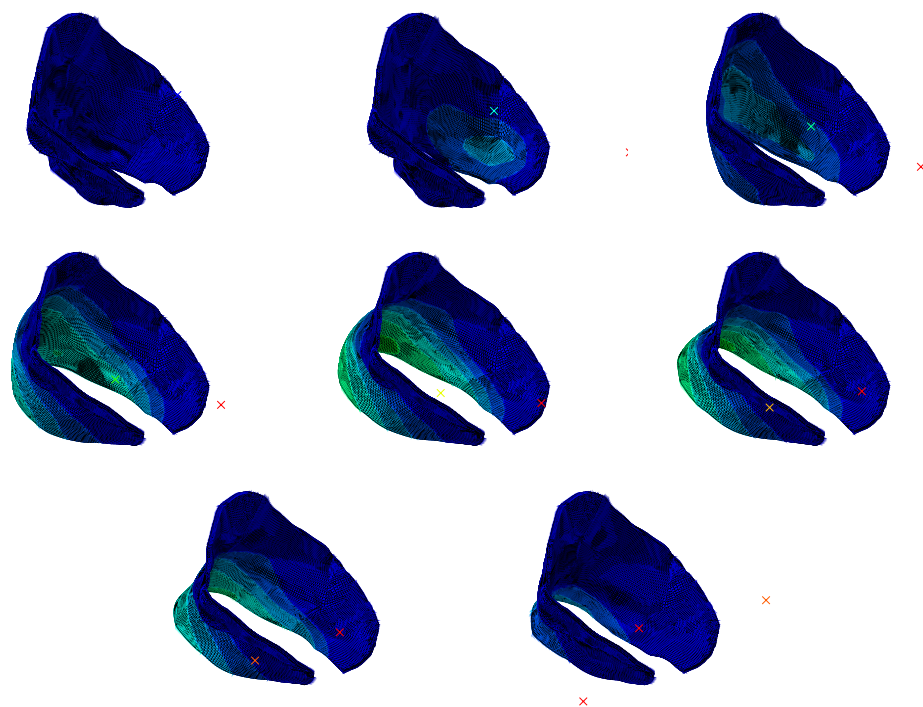


Figura 5.4: *Frames* exportados para o formato VRML correspondentes aos MPP, começando no *frame* 0 do *step* 0 (canto superior esquerdo) e terminando nos *frames* 0 e 803 do *step* 7 (imagens da última linha, correspondentes ao primeiro e último *frames* do último *step*).

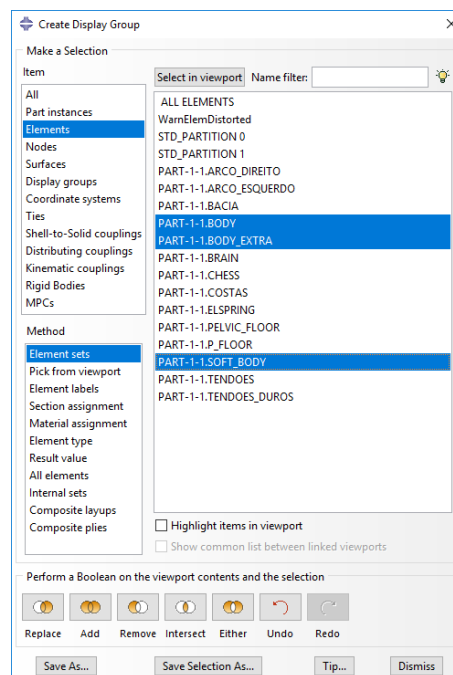


Figura 5.5: Janela de seleção de componentes a incluir no modelo (evidenciados encontram-se os elementos correspondentes ao feto)

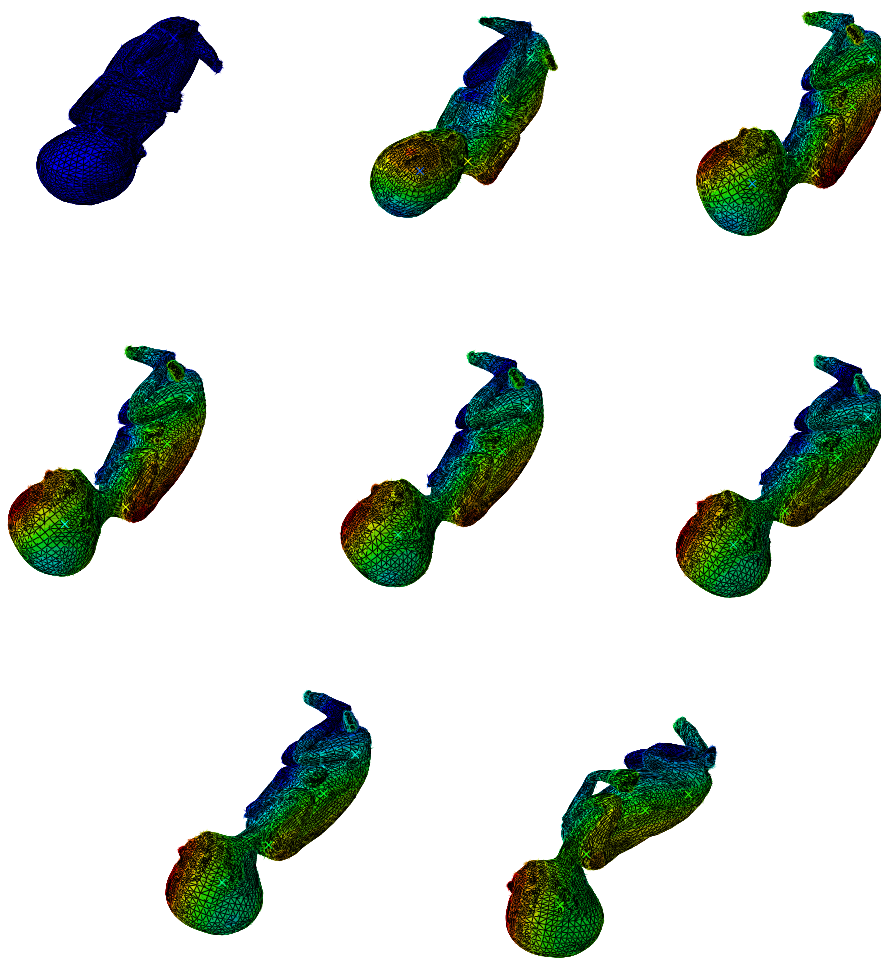


Figura 5.6: *Frames* exportados para o formato VRML correspondentes ao feto, começando no *frame* 0 do *step* 0 (canto superior esquerdo) e terminando nos *frames* 0 e 803 do *step* 7 (imagens da última linha, correspondentes ao primeiro e último *frames* do último *step*).

5.2 Conversão para o Formato FBX

Tendo todos os objetos desejados no formato VRML, o passo seguinte para atingir um formato compatível com realidade virtual é a conversão para ficheiros do tipo FBX, formato este reconhecido pelo Unity e que mantém as informações dos valores de deslocamento para cada nó e das cores correspondentes para as faces que compõem a superfície dos objetos.

Para conseguir então a criação de ficheiros FBX, é necessária a utilização do *software* Blender, que é capaz de reconhecer o formato inicial (VRML) e permite a exportação para o tipo FBX. A importação dos ficheiros VRML é realizada pelos comandos pré-definidos do Blender, ou seja, indo a “File » Import » X3D Extensible 3D (.x3d/.wrl)”.

Da mesma forma que foi referido no subcapítulo 5.1, este processo de conversão deveria ser automático, mas devido a complicações encontradas e referidas no capítulo 7.2, este passo foi realizado manualmente.

A conversão de VRML para FBX é bastante direta, quando os objetos pretendidos se encontram já completamente tratados e sem elementos desnecessários. Nas primeiras tentativas realizadas na presente dissertação este problema foi observado, conforme se pode observar na figura 5.7. Para ultrapassar esta questão, o processo anterior descrito no subcapítulo 5.1 teve que ser repetido, mas garantindo que antes de realizar a exportação para VRML a informação relativa à legenda e restantes informações se encontra desativada (tirando o visto em todas as caixas das opções denominadas de “Viewport Annotations” - ver figura 5.8) e identificando os restantes elementos desnecessários para o modelo.

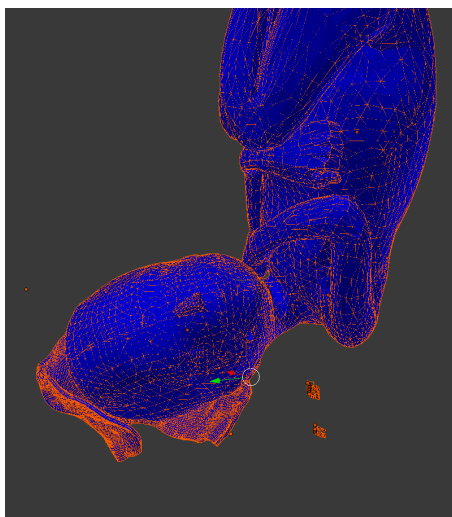


Figura 5.7: Modelo com elementos desnecessários. Imediatamente abaixo do corpo do feto podem-se observar alguns componentes da escala do modelo e alguns elementos de texto.

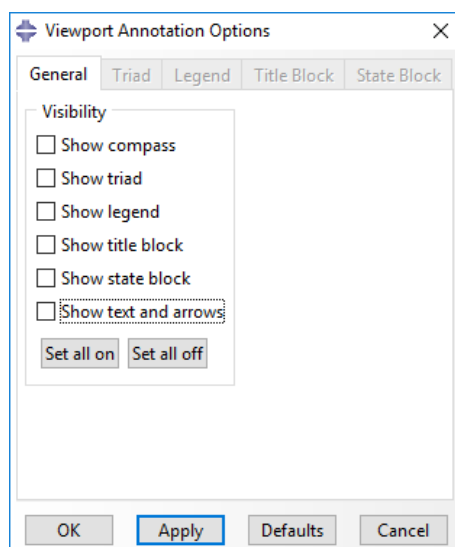


Figura 5.8: Janela de seleção de elementos extra do modelo. Esta área encontra-se em “Viewport » Viewport Annotation Options...”.

Ultrapassando este problema, o objeto (ver figura 5.9) encontra-se então pronto para exportação para o formato FBX. Deve-se apenas verificar se tudo se encontra correto. Tendo em conta que os objetos utilizados na presente dissertação possuem um tamanho bastante elevado (o que por vezes induz o utilizador em erro, pois quando a importação do VRML é realizada a janela aparenta não ter resultado em nada), é necessário diminuir consideravelmente a escala de visualização.

Caso todo o objeto esteja correto, basta que o utilizador exporte tudo para a extensão FBX através da funcionalidade “File » Export » FBX (.fbx)”. Este processo, conforme referido, foi realizado manualmente, ou seja, importando e exportando cada *frame* individualmente.

No fim da exportação de todos os *frames* dos MPP e do feto (dezasseis no total), é possível a criação da aplicação no *software* Unity.

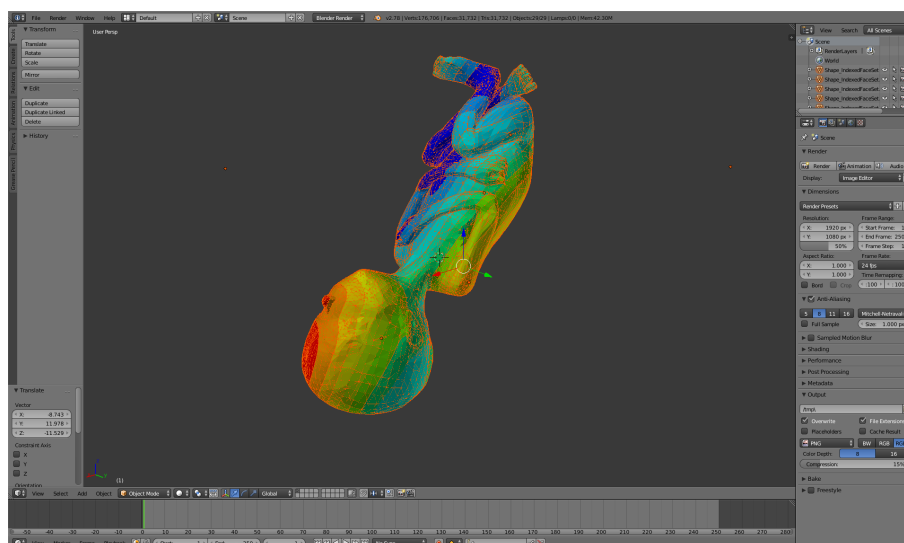


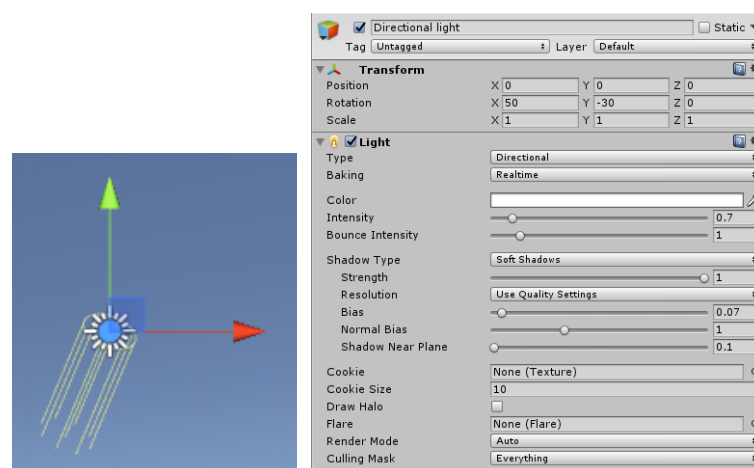
Figura 5.9: Objeto completamente processado e pronto para exportação para o formato FBX.

5.3 Construção da Aplicação no Unity - Importação do Modelo para Realidade Virtual

Após a conversão de todos os objetos pretendidos (oito *frames* para os MPP e oito *frames* para o feto) para o formato FBX, o passo seguinte no desenvolvimento da aplicação passa pela construção do ambiente virtual contendo o modelo biomecânico do parto. Desta forma, a primeira etapa consiste na disposição dos elementos obrigatórios da chamada cena no *software* Unity. Para que um determinado mundo virtual possa ser visualizado em realidade virtual, é necessário colocar componentes que simulem iluminação, caso contrário qualquer objeto pertencente à cena parecerá escuro (como se o utilizador se encontrasse numa sala sem iluminação). É também obrigatório incluir algo que determine o ponto de vista do utilizador, caso contrário o utilizador pode não conseguir visualizar nenhum elemento pretendido, por (virtualmente) se encontrar noutra local do mundo.

O Unity contém nas suas pré-definições a capacidade de introduzir elementos-padrão numa cena virtual, chamados de *GameObjects* (ou Objetos de Jogo), onde se podem encontrar vários elementos que simulam pontos de luz de diversos tipos diferentes (luz direcional, ponto de luz, entre outros). Entre estes *GameObjects* pode-se encontrar também uma câmara que, quando posicionada, imita o ponto de vista do utilizador (ou seja, dependendo da posição e orientação da câmara, o utilizador consegue visualizar o local para onde a câmara está a apontar).

De forma a facilitar o processo de inclusão dos *frames* referidos no subcapítulo 5.2, optou-se por colocar tanto a iluminação (o tipo escolhido para a presente dissertação foi uma fonte de luz direcional de intensidade 0.7 unidades de 8.0 unidades possíveis, com uma rotação de 50° no eixo X, -30° no eixo Y e 0° no eixo Z - ver figura 5.10) como a câmara no ponto (0, 0, 0) em unidades do mundo virtual. É importante referir que o Unity possui como referencial pré-definido o que se encontra ilustrado na figura 5.11.



(a) Ponto de luz direcional inserido no Unity (b) Coordenadas e valor de intensidade do ponto de luz direcional

Figura 5.10: Características do ponto de luz inserido no Unity



Figura 5.11: Sistema de coordenadas do Unity

De modo a poder visualizar o mundo criado (que até este ponto não contém nenhum objeto, apenas contém um mundo aparentemente infinito iluminado a partir do ponto de vista do utilizador), é necessária a configuração do projeto para os óculos de realidade virtual utilizados na presente dissertação. Para atingir este objetivo, em “Edit » Project Settings » Player” é possível marcar a caixa correspondente à configuração “Virtual Reality Supported” e definir o *hardware* desejado, que no caso da presente dissertação se trata dos Oculus Rift.

Concluídos estes processos, o projeto encontra-se pronto para iniciar a introdução dos *frames* em FBX. Estes devem ser colocados dentro da pasta do projeto, para que o Unity consiga aceder a eles mais facilmente e importá-los para a biblioteca de recursos.

Para iniciar a introdução dos objetos na cena virtual, e proceder à configuração dos posicionamentos e escalamentos dos elementos, optou-se por incluir em primeiro lugar o sistema esquelético que servirá apenas para auxiliar a visualização da simulação do parto e para dar ao utilizar alguma referência. O conjunto dos ossos retrata apenas a pelve, e inclui ainda uma porção dos fêmures e da coluna vertebral, conforme se pode observar na imagem 5.12. Este objeto foi utilizado a partir de estudos já realizados anteriormente.

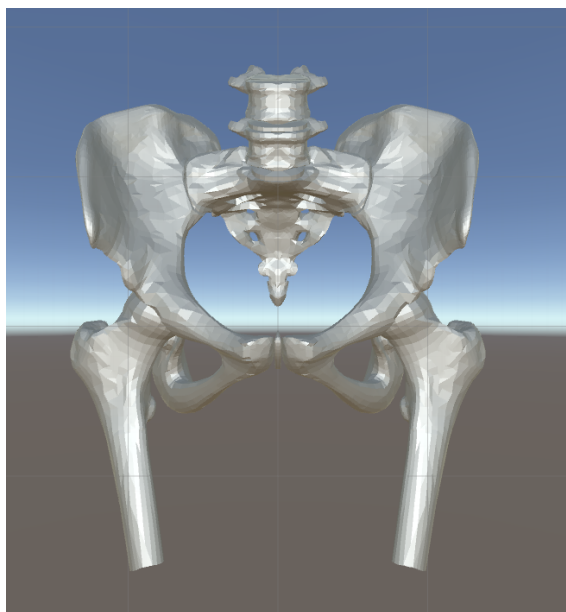


Figura 5.12: Sistema esquelético da pelve inserido no Unity

Ao introduzir o modelo observa-se que este não se encontra numa escala apropriada para ser visível pelo utilizador do ponto de vista escolhido, uma vez que o sistema possui um tamanho consideravelmente maior. Desta forma, por tentativa e erro, procurou-se um fator de escala adequado para que todo o modelo pudesse ser facilmente analisado pelo utilizador, tendo-se optado pelo fator de escala de 0.003 (em relação ao tamanho inicial) nos três eixos, como se pode observar na figura 5.13.

Além disso, escolheu-se também modificar a posição e rotação do modelo, a primeira para que a simulação decorresse ao nível dos olhos do utilizador, e a segunda apenas por motivos mais estéticos (ver figura 5.13).

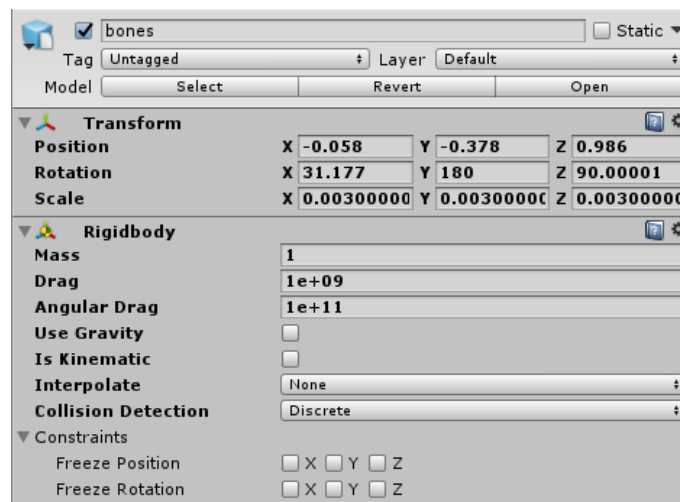


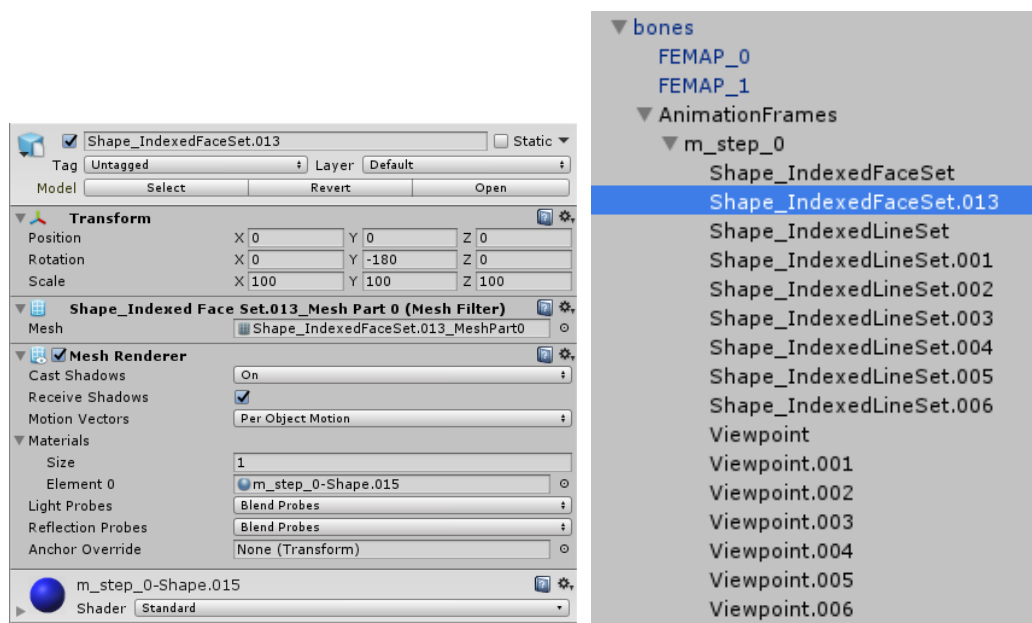
Figura 5.13: Posição, orientação e escalamento do sistema esquelético no Unity (*GameObject* com o nome “bones”)

De seguida, e como forma de validação para todos os *frames* referidos anteriormente, escolheu-se o *frame* 0 do primeiro *step* dos MPP e foi-lhe aplicado o mesmo fator de escala e valores de translação e rotação, de forma a verificar se coincidiam com o sistema esquelético. Para incluir este objeto, o Unity permite que um ficheiro do tipo FBX seja simplesmente arrastado para a cena criada, pelo que basta que o utilizador arraste o ficheiro para a área existente do lado esquerdo da interface do Unity, correspondente à “Hierarchy”, processo que cria automaticamente um *GameObject* que contém todas as informações pertencentes ao *frame* desejado, que no caso do modelo biomecânico do parto consiste na informação do deslocamento de cada nó da superfície dos MPP.

Após a aplicação da posição, rotação e escalamento, verifica-se que o esqueleto e o *frame* introduzido não coincidem, pelo que foi necessário identificar os valores corretos para que ambos se adaptassem um ao outro. Utilizando novamente o método de tentativa e erro, e através da observação, chegou-se à conclusão de que os valores têm que ser personalizados para os objetos referentes aos MPP e ao feto.

Uma vez que durante o desenvolvimento do projeto se sentiu necessidade de introduzir *scripts*, e que as características de um *GameObject* são transmitidas à medida que a hierarquia desce (ou

seja, as características de um objeto-pai são transmitidas para o objeto-filho que lhe está associado), definiram-se os valores representados na figura 5.14 para cada ficheiro FBX, e que são afetados pelos valores aplicados ao *GameObject* correspondente ao sistema esquelético (valores esses visíveis na figura 5.13). Na figura 5.15 é possível também visualizar a hierarquia existente entre os objetos correspondentes ao sistema esquelético e aos *frames* do modelo biomecânico.



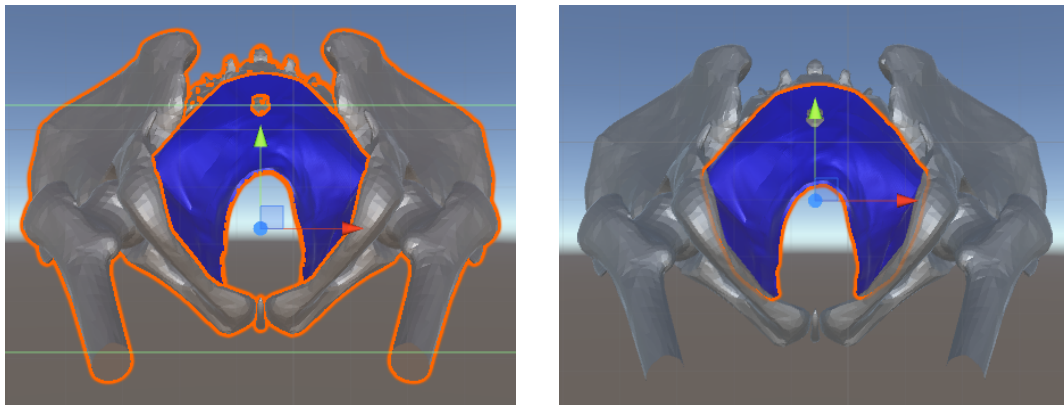
(a) Posição, orientação e escalamento de um dos *frames* dos MPP (*GameObject* com o nome “Shape_IndexedFaceSet.013”)

(b) Hierarquia do *GameObject* “bones”, contendo o *GameObject* “Shape_IndexedFaceSet.013”. Os valores de rotação e escalamento aplicados aos objetos das hierarquias mais baixas são afetados pelos valores aplicados aos que os contêm.

Figura 5.14: *GameObject* “Shape_IndexedFaceSet.013”

Concluída a calibração entre o sistema esquelético e o modelo FBX do *frame* 0 dos MPP (figura 5.16), pode-se concluir que os restantes quinze *frames* (referentes aos MPP e ao feto) deverão possuir os mesmos valores de translação, rotação e escalamento para que se adaptem ao modelo já criado.

Antes de se poder incluir o *frame* seguinte, é agora necessário criar um mecanismo que permita a animação do modelo, ou seja, é necessário que os *frames* sejam percorridos pelo Unity de forma ordenada e consecutiva, da mesma forma que um filme é constituído por uma sucessão de imagens. Este mecanismo apenas pode ser atingido com o auxílio de um *script* associado a um objeto que contenha numa hierarquia mais baixa que todos os *frames* que se pretendam animar, tornando o objeto seguinte visível e desativando o atual.



(a) *GameObject* “bones” selecionado. Como se pode observar, tanto o sistema esquelético como os MPP fazem parte deste objeto, sendo que os MPP são um *GameObject* pertencente a ele.

(b) *GameObject* “Shape_IndexedFaceSet.013” selecionado. Este objeto contém apenas o *frame* 0 do *step* 0 da simulação dos MPP.

Figura 5.15: Seleção dos *GameObjects*, demonstrando a hierarquia dos mesmos. É também possível visualizar o resultado da calibração entre o sistema esquelético e um dos *frames* do modelo biomecânico.

De modo a desempenhar esta função, foi desenvolvido um *script* em C# (linguagem de programação orientada a objetos), que de forma automática deteta o número de *frames* existentes nos objetos-filhos do objeto a que o *script* se encontra associado, e os percorre de forma ordenada, ativando um a seguir ao outro (e desativando o anterior para que não ocorra sobreposição de dois *frames*).

Este ficheiro de código encontra-se associado ao *GameObject* “AnimationFrames”, e o mesmo pode ser consultado no anexo A. A principal função que executa esta ação possui o cabeçalho “void Start()”, como se pode observar no anexo, em primeiro lugar é obtido o número de *frames* existente na hierarquia abaixo do objeto ao qual se associa este o *script* (ver código 5.1).

Código 5.1: Comando responsável pela contagem de *frames*

```
frames = new Transform[this.transform.childCount];
```

Posteriormente, esta variável (que agora contém uma lista de todos os *frames* existentes) é percorrida dentro de um ciclo *for* até chegar ao último *frame*, e repetindo novamente todo o ciclo em *loop* (ver porção de código 5.2).

Código 5.2: Ciclo que percorre todos os *frames* existentes

```
for(int i = 0; i < frames.Length; i++)
{
    frames[i] = this.transform.GetChild(i);
    frames[i].gameObject.SetActive(false);
}
```

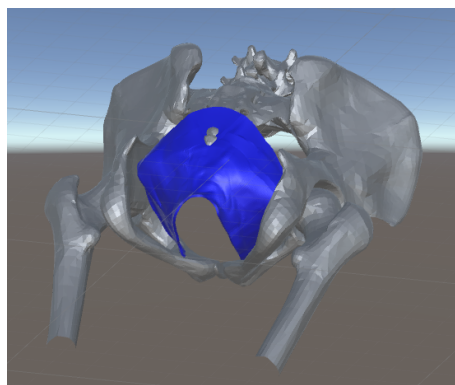


Figura 5.16: Exemplo do modelo biomecânico inserido no ambiente virtual. No caso retratado observa-se o *frame* 0 do *step* 0 da simulação.

Solucionada esta questão, basta agora acrescentar todos os restantes *frames* ao modelo, tendo apenas que garantir que somente o primeiro se encontra ativo (característica possível de se confirmar através do visto retratado na figura 5.17).

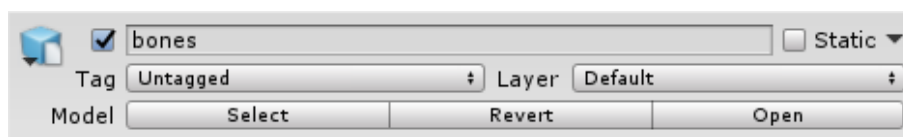


Figura 5.17: Ativação ou desativação de um *GameObject* no Unity. Caso a caixa existente no lado esquerdo do nome do objeto possua um visto, o *GameObject* encontra-se ativo.

O ambiente virtual, após incluir todos os *frames* dos MPP e do feto, resultou na hierarquia de *GameObjects* visível na figura 5.18, e no objeto retratado na figura 5.19.

É de notar que com a ajuda do *script*, pressionando o botão de “Play” do Unity e entrando em modo de “Jogo”, o utilizador consegue visualizar a animação do modelo biomecânico do parto, *frame* a *frame*, sendo que quando o último é alcançado, o *script* referido implementa uma função de “Loop” que, como o nome indica, leva o modelo a voltar ao primeiro *frame*.

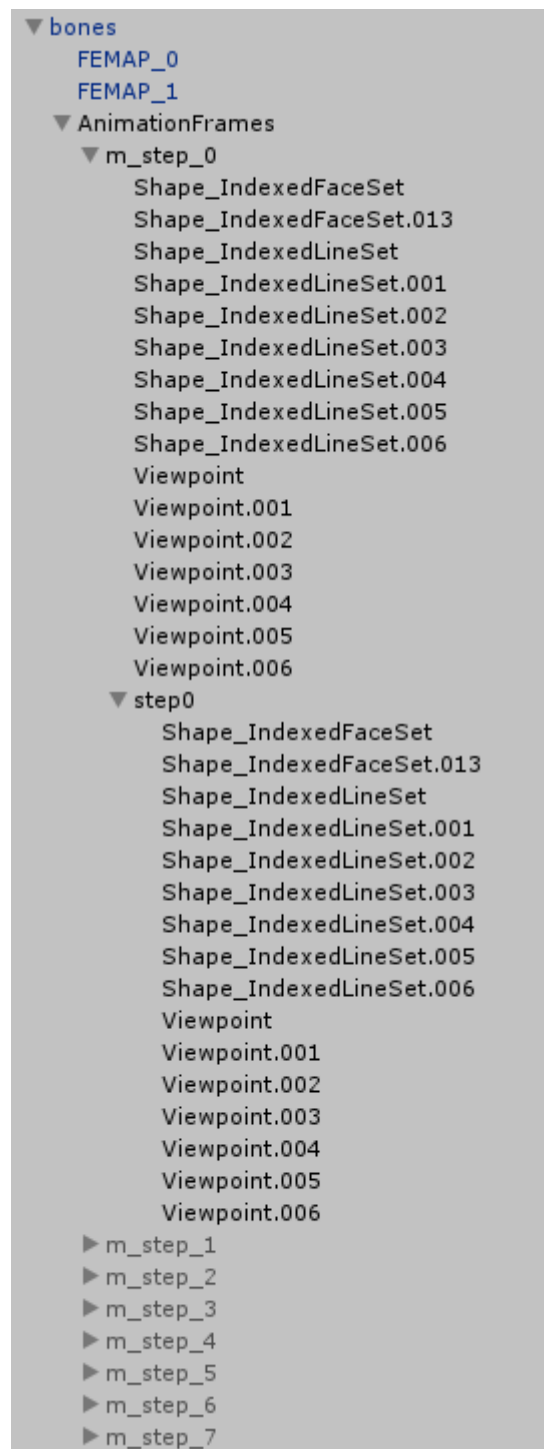


Figura 5.18: Hierarquia final do *GameObject* “bones”. Os objetos “FEMAP_0” e “FEMAP_1” correspondem ao sistema esquelético da pelve. O objeto “AnimationFrames” possui associado a si o código em C# referido e agrega os *frames* do modelo biomecânico, para simplificar a modificação do mesmo. Os objetos “m_step_X” (com X entre 0 e 7) correspondem aos 8 *frames* dos MPP, enquanto que os objetos “stepY” (com Y entre 0 e 7) correspondem ao feto.

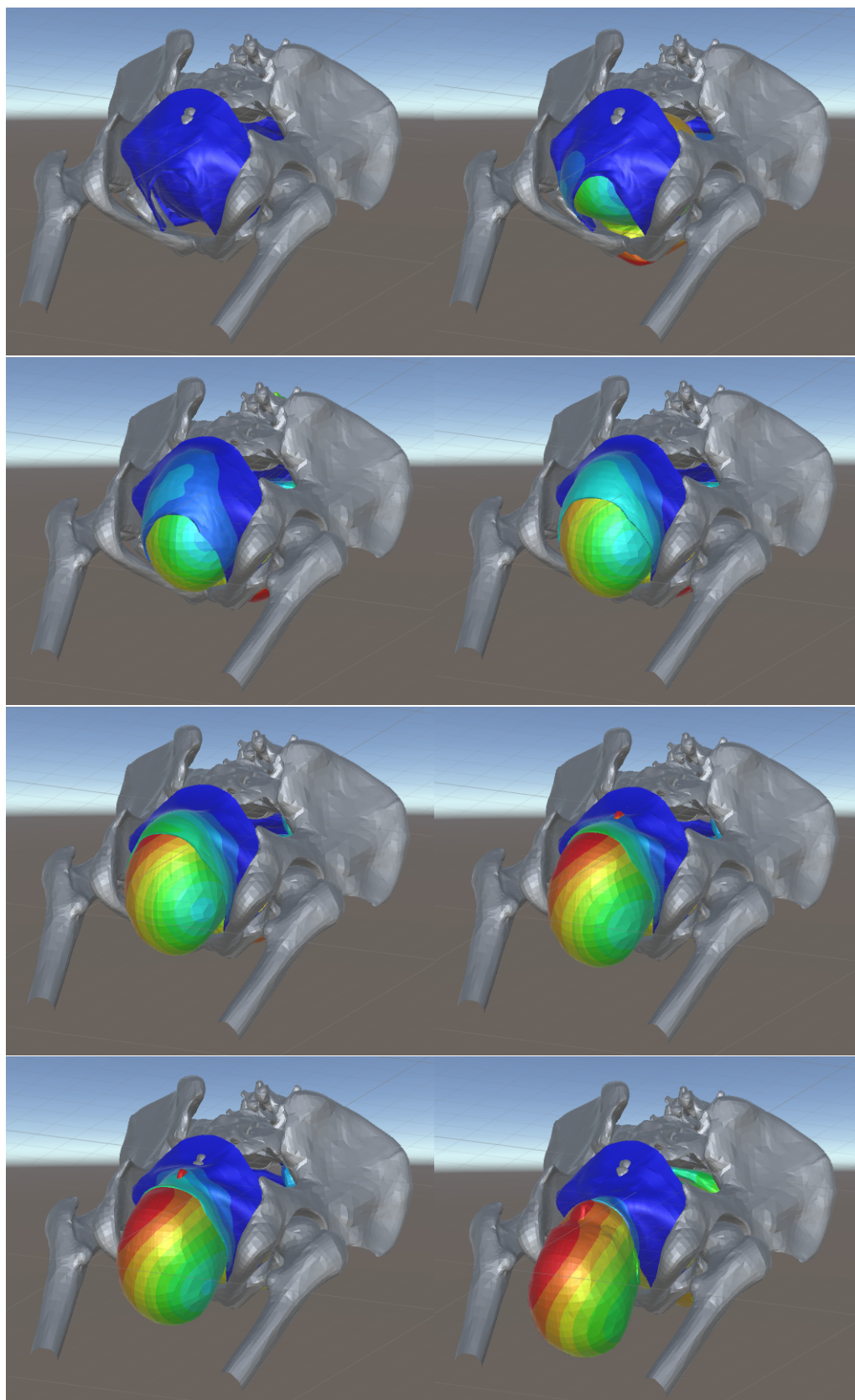


Figura 5.19: Modelo biomecânico final inserido no Unity, com os oito *frames* dos MPP e do feto

Capítulo 6

Interface com o Utilizador

Terminadas as etapas referidas no capítulo 5, tem-se como resultado uma animação na interface do Unity que contém uma sucessão de *frames* que ilustram o processo de um parto humano através do canal vaginal, de um ponto de vista biomecânico.

No entanto, a única funcionalidade existente é a visualização do modelo através do Oculus Rift, dando ao utilizador a imersividade que lhe permite observar (de vários ângulos) os pontos com níveis de deslocamento mais elevados nos MPP e no feto.

Deste modo, decidiu-se implementar algumas ações que permitam ao utilizador uma interação com o modelo biomecânico em realidade virtual mais completa. Nos próximos subcapítulos irá ser abordada a interação com o modelo através da criação de um menu virtual e da associação com o dispositivo Leap Motion.

6.1 Introdução do Dispositivo Leap Motion

Antes da criação de um menu que possua novas funcionalidades, é necessário configurar o dispositivo capaz de detetar as mãos do utilizador e introduzi-las no mundo virtual.

Graças à existência do SDK e da API já fornecidos pela empresa detentora do dispositivo em [69], a inclusão do Leap Motion no Unity é facilmente estabelecida, uma vez que os *plugins* de compatibilidade se encontram já incluídos. Para terminar a configuração, foram apenas seguidos os passos sugeridos pelo *website* do produto, disponíveis em [70].

Terminada a configuração, e como pode ser consultado em [69] é necessária a introdução dos *GameObjects* que estejam associados a cada uma das mãos do utilizador, e que possuam os *scripts* que permitam o correto funcionamento do dispositivo. Desta forma, foi em primeiro lugar inserido um objeto pré-fabricado designado de “LMHeadMountedRig” que contém na sua hierarquia um *GameObject* chamado “LeapHandController”, e que é responsável pela definição do volume dentro do qual as mãos do utilizador são detetadas (ou seja, qual é o volume em frente do dispositivo que é utilizado como espaço útil para a deteção das mãos, e que pode ser configurado), conforme se pode observar na figura 6.1.

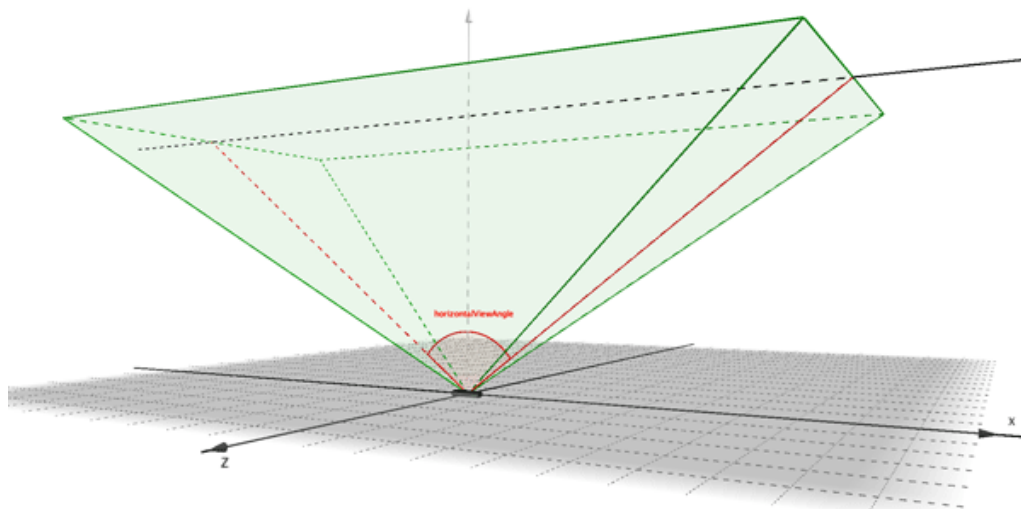


Figura 6.1: Volume dentro do qual o Leap Motion consegue detetar objetos. Na imagem é possível observar que os eixos pré-definidos pelo Leap Motion não correspondem aos eixos do Unity, pelo que é necessário um ajuste a nível do código. (retirado de [34])

De seguida, é necessário incluir como objetos-filhos do “LeapHandController” as mãos virtuais que o utilizador verá no lugar das suas quando a simulação estiver a funcionar, arrastando para o Unity os modelos de mãos já fornecidos pelo Leap Motion (figura 6.2). No caso da presente dissertação, foi escolhido o modelo “RigidRoundHand”.

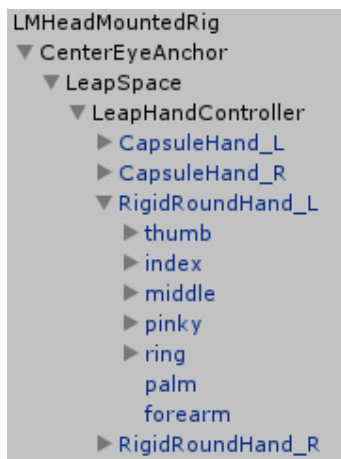


Figura 6.2: Hierarquia da *GameObject* “LMHeadMountedRig”, contendo o objeto responsável pela definição do volume de deteção do Leap Motion, “LeapHandController”, e os objetos correspondentes às mãos e dedos virtuais.

Após garantir que todas as recomendações sugeridas em [69] foram seguidas, e colocando a simulação no Unity a correr, é possível observar que as mãos do utilizador já se encontram no ambiente virtual do modelo biomecânico do parto (ver figura 6.3), pelo que os passos seguintes consistem na determinação das funcionalidades a implementar e na criação do menu já referido.

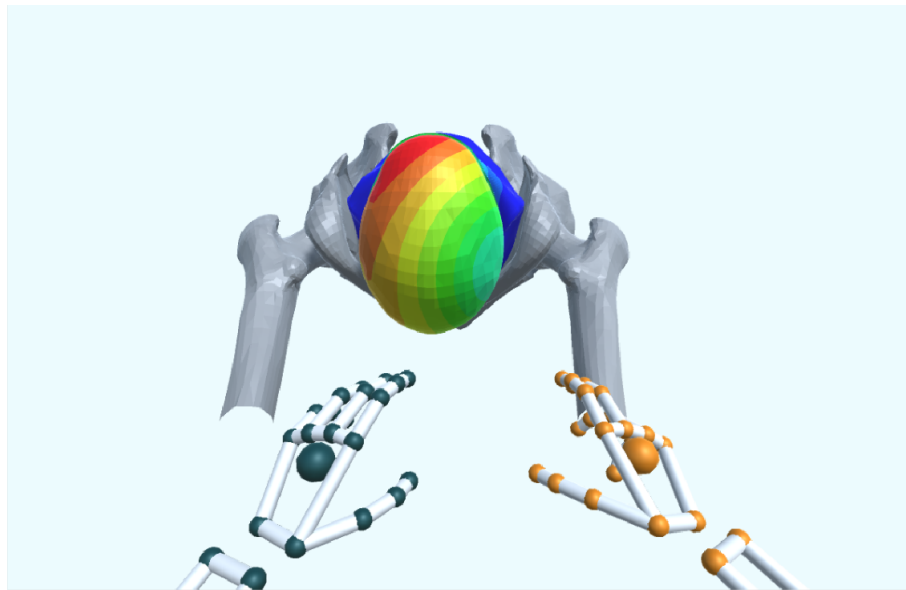


Figura 6.3: Modelo “RigidRoundHand” para as mãos utilizado na simulação

6.2 Determinação das Funcionalidades a Implementar

Antes da criação do menu, é necessário decidir quais serão as novas funcionalidades a incluir no modelo que permitam uma interação entre este e o utilizador.

Uma vez que a simulação em realidade virtual se trata de uma sequência de *frames* até ao momento, as funcionalidades apenas poderão incluir ações relacionadas com animações. Deste modo, optou-se por implementar as seguintes funções:

- **Play/Pausa** - Esta função tem como objetivo pausar a sequência de animação no *frame* em que a mesma é ativada ou retornar à animação a partir do *frame* em que se encontra parado.
- **Avançar/Retroceder *Frame*** - Quando a animação se encontrar pausada, deverá ser possível que o utilizador avance ou retroceda um só *frame* de cada vez, de modo a que possa escolher aquele que deseja.
- **Escolha do *Frame*** - Como alternativa à função anterior, deverá existir uma forma de selecionar diretamente o *frame* desejado pelo utilizador sem ter que percorrer todos os existentes.
- **Rotação do Modelo** - Esta função permite que o utilizador rode o modelo em torno dos eixos X e Y, de forma a colocar o objeto na posição que forneça ao utilizador novos pontos de vista da simulação.
- **Velocidade da Animação** - O utilizador pode escolher uma velocidade específica na sequência de *frames*, fazendo com que a animação ocorra de forma mais rápida ou lenta.
- **Translação do Modelo** - Através de um gesto com a mão do utilizador, deverá ser possível a translação do modelo para uma localização escolhida, arrastando-o com a mão.

- **Ativação/Desativação do Menu** - Com um gesto da mão do utilizador, o menu que serve de interface e permite a execução das funções descritas pode surgir em frente do utilizador ou ser desativado.

Mais funções poderiam ser implementadas na presente dissertação, mas como será explicado no subcapítulo 7.2 optou-se por incluir apenas as funções referidas nesta versão do projeto.

6.3 Implementação das Funcionalidades

Para realizar as funções referidas, optou-se pela criação de um menu, colocado em frente do modelo biomecânico, utilizando como *trigger* das funções os botões e *sliders* já existentes no Unity. Estes são *GameObjects* pré-definidos na interface do *software*, e que podem ser personalizados pelo utilizador para cumprir as funções desejadas.

Desta forma, introduziu-se um objeto retangular que serve como base do menu, onde irão ser colocados os botões e *sliders* com as funções referidas 6.4a. Seguidamente foram introduzidos os respetivos objetos que irão desempenhar as funcionalidades, tendo-se optado pela disposição observável na figura 6.4b. Foram também incluídos alguns sons, de modo a dar indicação ao utilizador de que o botão foi pressionado. Na figura 6.5 encontram-se descritos quais os botões e *sliders* que respondem a cada uma das funcionalidades implementadas.

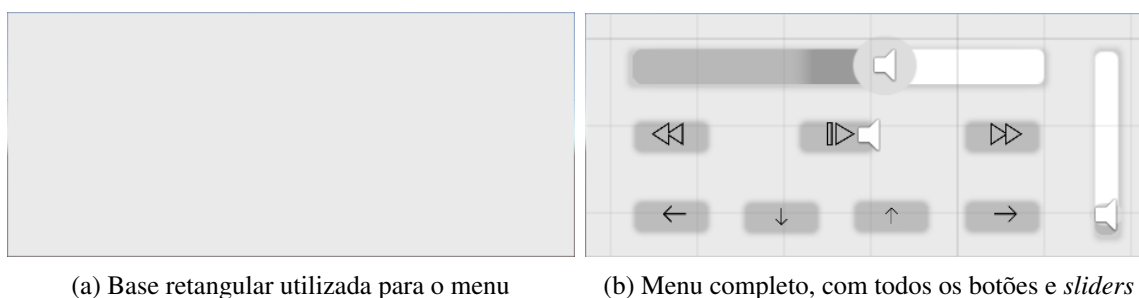


Figura 6.4: Evolução do desenvolvimento do menu na simulação

Possuindo os elementos gráficos concluídos, resta então implementar as funções propriamente ditas. Para atingir este objetivo, foi criado um novo *script* em C# que, em conjunto com o ficheiro de código já criado e que pode ser consultado no anexo A, descreve todas as ações que devem ocorrer no modelo quando um dos botões é ativado.

Usando as funcionalidades já pré-definidas no *software* utilizado para edição de código (Microsoft Visual Studio) de compatibilidade com o Unity, o desenvolvimento de funções é simplificada. O *script* desenvolvido pode ser consultado no anexo B. Deve-se ter em conta que algumas das funções chamadas por este ficheiro se encontram descritas no anexo A, pelo que os dois ficheiros de código devem ser colocados no mesmo diretório.

No caso da funcionalidade referente à rotação do modelo nos eixos X e Y, foi incluída no ficheiro de código a possibilidade de se usar as setas do teclado como *input*, como se pode observar na porção de código 6.1.

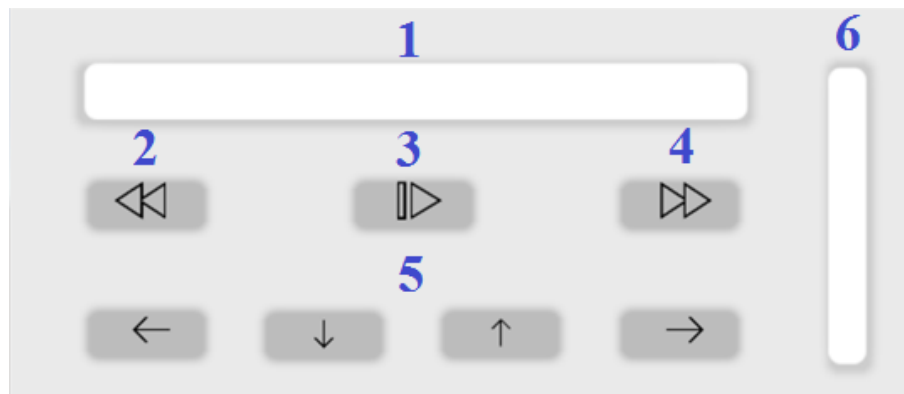


Figura 6.5: Botões e *sliders* do menu. 1- Escolha do *frame* quando animação está pausada; 2- Retroceder um *frame* quando animação está pausada; 3- Play/Pausa da animação; 4- Avançar um *frame* quando animação está pausada; 5- Rotação do modelo nos eixos X e Y; 6- Escolha da velocidade da animação.

Código 6.1: Definição das funções de rotação do modelo biomecânico

```

if (Input.GetKey(KeyCode.RightArrow) || (right == true &&
    rRight.enabled == true))
{
    gameObject.transform.RotateAround(meshesCenter, Vector3.up,
        speed * Time.deltaTime);
}
else if (Input.GetKey(KeyCode.LeftArrow) || (left == true &&
    rLeft.enabled == true))
{
    gameObject.transform.RotateAround(meshesCenter, Vector3.up,
        -speed * Time.deltaTime);
}
else if (Input.GetKey(KeyCode.UpArrow) || (up == true &&
    rUp.enabled == true))
{
    gameObject.transform.RotateAround(meshesCenter, Vector3.right,
        speed * Time.deltaTime);
}
else if (Input.GetKey(KeyCode.DownArrow) || (down == true &&
    rDown.enabled == true))
{
    gameObject.transform.RotateAround(meshesCenter, Vector3.right,
        -speed * Time.deltaTime);
}

```


Associando este *script* ao *GameObject* referente ao menu, as funções de correr/pausar animação (função “public void Play()” do ficheiro “FrameByFrameAnimation” - anexo A), retroceder/avançar *frame* quando a animação está pausada (funções “public void GoBackwards()” e “public void GoForward()” do ficheiro “FrameByFrameAnimation”), escolha do *frame* desejado quando a simulação está pausada (função “public void SliderPosition(float percentagem)” que recebe como parâmetro do menu no Unity a percentagem a que o indicador se encontra), a escolha da velocidade da animação (função “public void SliderSpeed(float percentagem)” que recebe a percentagem a posição do *slider* correspondente a esta função, e inicializando na percentagem 8% da velocidade máxima permitida - 60 fps), e a rotação do modelo (funções “public void rotateLeft()”, “public void rotateRight()”, “public void rotateUp()” e “public void rotateDown()”) encontram-se assim definidas e a funcionar como esperado no modelo. Note-se que para que estas funções sejam aplicadas ao modelo, é necessário associar o *script* “ModelRotation” - anexo B - ao *GameObject* “bones”, para que as consequências das funcionalidades ocorram no modelo biomecânico inteiro, como se pode observar na figura 6.6.

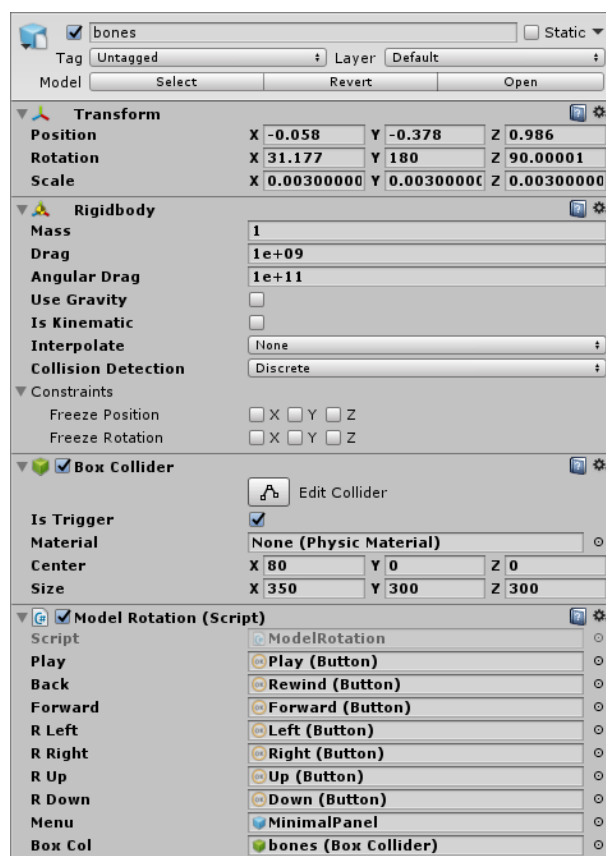


Figura 6.6: Descrição do *GameObject* “bones”, com a associação do *script* “ModelRotation” e da caixa de colisão.

Além destas, outra das funcionalidades ainda não implementada é a ativação ou desativação do menu. Para se inserir esta função no modelo, pensou-se em recorrer à API fornecida pela Leap Motion, nomeadamente qual foi a primeira mão a surgir no campo de deteção do dispositivo, bem

como na diferenciação da detecção entre os diferentes dedos da mão. As funções que implementam estas distinções entre os segmentos corporais encontram-se descritas no código 6.2.

Código 6.2: Identificação da primeira mão a ser detetada pelo Leap Motion e determinação da posição no eixo X dos dedos polegar e mindinho.

```
if (frame.Hands.Count > 0)
{
    Hand primeHand = frame.Hands[0];
    foreach (Finger f in primeHand.Fingers)
    {
        if (f.Type == Finger.FingerType.TYPE_THUMB)
        {
            thumbPos = f.StabilizedTipPosition.x;
        }
        if (f.Type == Finger.FingerType.TYPE_PINKY)
        {
            pinkyPos = f.StabilizedTipPosition.x;
        }
    }

    [...]
}
```

Como se pode observar, a primeira mão a ser detetada é colocada na variável “frame.Hands[0]” e para detetar qual das duas mãos foi a primeira a surgir do campo do Leap Motion, optou-se por identificar a posição no eixo X do dedo mindinho (“Finger.FingerType.TYPE_PINKY”) e do polegar (“Finger.FingerType.TYPE_THUMB”).

Após obter estes valores, o raciocínio utilizado consistiu no seguinte: se o dedo mais à esquerda for o polegar, então a primeira mão a ser detetada foi a mão esquerda; caso o dedo mais à esquerda for o mindinho, então a primeira mão foi a mão direita.

Sabendo qual a primeira mão a aparecer no espaço de detecção, definiu-se que o menu seria ativado ou desativado mediante a rotação do pulso no eixo longitudinal do antebraço, sendo que o mesmo seria ativado quando a palma da mão se encontra voltada para cima, ou seja quando ocorre o movimento de supinação, e desativado quando a mesma se encontra voltada para baixo, ou seja, quando ocorre movimento de pronação (ver figura 6.7).

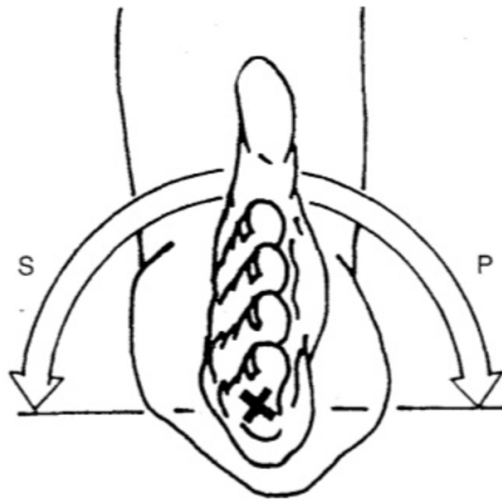


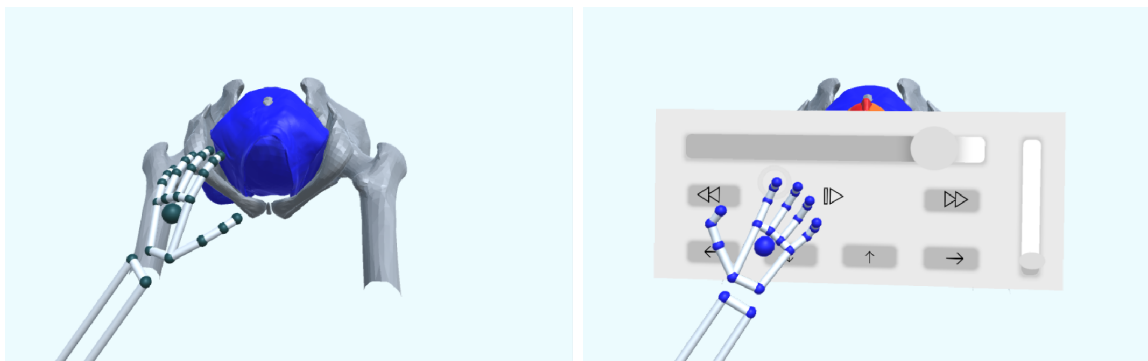
Figura 6.7: Movimentos de supinação e pronação do pulso. O S na figura corresponde ao movimento de supinação e o P ao movimento de pronação. (retirado de [71])

Código 6.3: Ativação ou desativação do menu consoante a rotação da primeira mão detetada

```
if (primeHand.IsLeft)
{
    if (thumbPos < pinkyPos)
    {
        boxCol.enabled = false;
        menu.SetActive(true);
    }
    else
    {
        boxCol.enabled = true;
        menu.SetActive(false);
    }
}
if (primeHand.IsRight)
{
    if (thumbPos > pinkyPos)
    {
        boxCol.enabled = false;
        menu.SetActive(true);
    }
    else
```

```
{  
    boxCol.enabled = true;  
    menu.SetActive(false);  
}  
}
```

Como se pode observar na porção de código 6.3, quando a mão esquerda é detetada como sendo a primeira mão no espaço de deteção e esta se encontra voltada para cima (ou seja, quando a posição no eixo X do polegar - variável “thumbPos” - é inferior ao valor da coordenada do mindinho - variável “pinkyPos”, o que significa que a palma esquerda se encontra virada para cima), então o menu é ativado (comando “menu.SetActive(true)”). Quando a palma da mão se encontra voltada para baixo, então o menu é desativado (comando “menu.SetActive(false)”). Esta funcionalidade (já implementada) pode ser observada na figura 6.8. A característica de colisão entre a mão do utilizador e os diferentes botões já se encontra pré-definida, pelo que não é necessária nenhuma outra configuração.



(a) Menu desativado através do movimento de pronação do pulso.

(b) Menu ativado através do movimento de supinação do pulso.

Figura 6.8: Implementação da funcionalidade de ativação/desativação do menu. No caso retratado, a mão que controla a função é a mão esquerda.

De modo a implementar a função que resta, a translação do modelo, e uma vez que a colisão entre o utilizador e o modelo (dentro do mundo virtual) não está definida, foi introduzida uma caixa de colisão em redor do *GameObject* “bones” através do componente “Box Collider” visível na figura 6.6. Para esta caixa, invisível para o utilizador, foi dado como centro o ponto (80, 0, 0), que como já foi referido se encontra afetado de uma rotação e um escalamento aplicados ao objeto “bones”, e foi ainda definido o seu tamanho (ver figura 6.9).

De modo a movimentar o modelo para o local pretendido pelo utilizador, durante a animação, basta que o utilizador tenha o menu desativado e que realize o gesto com uma das mãos de agarrar (fechar os dedos) a caixa de colisão, e movimente o modelo mantendo a mão fechada.

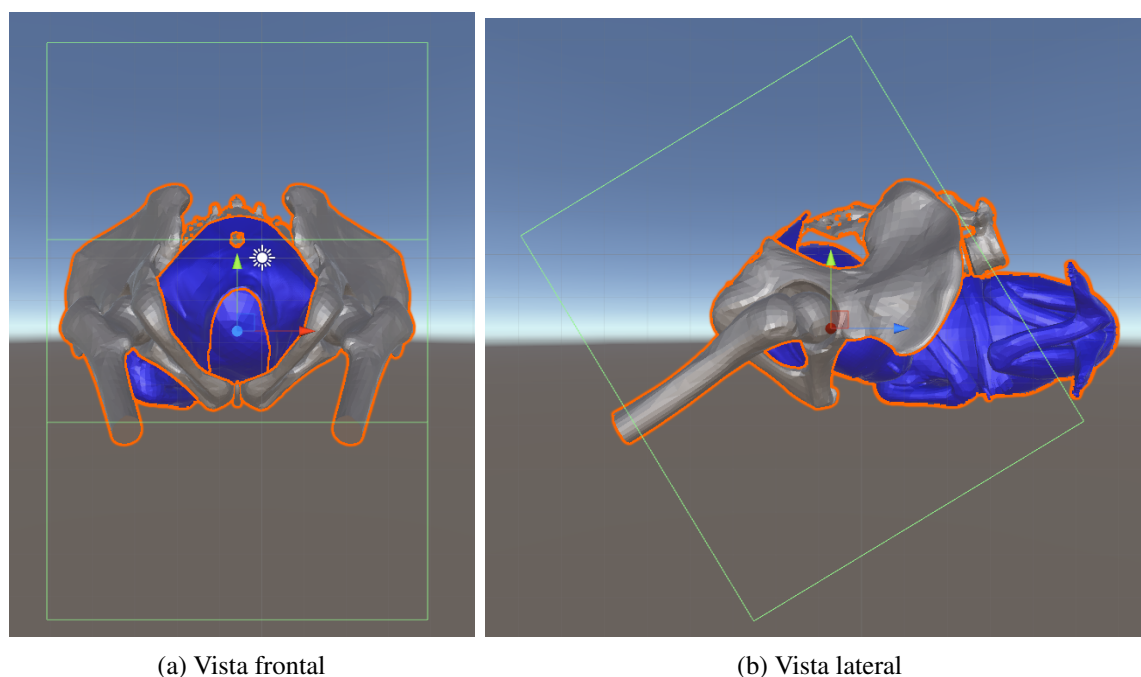


Figura 6.9: Caixa de colisão (a verde) associada ao *GameObject* “bones”

Além disso, e como é visível no código 6.3, esta caixa de colisão é ativada e desativada inversamente ao menu, ou seja, quando o menu é ativado, a caixa de colisão é desativada, e vice-versa. Esta particularidade foi definida para que o utilizador não corra o risco de mover o modelo quando se encontra a carregar num dos botões do menu.

6.4 Criação do Executável

Terminada a implementação da interatividade entre o utilizador e o modelo biomecânico importado para realidade virtual, e face aos objetivos definidos no início da presente dissertação, resta apenas um passo: a exportação de todo o modelo, incluindo a visualização com os Oculus Rift, o menu e a interação com o Leap Motion, para um formato que possa ser utilizado em qualquer computador que suporte realidade virtual (em termos das especificações de placa gráfica e processamento do próprio computador).

O Unity permite a exportação da cena criada para diferentes ambientes, entre os quais Android, Xbox One, entre outros. A opção pertinente para a presente dissertação é referida como “PC, Mac & Linux Standalone”, e encontra-se em “File » Build Settings...” (ver figura 6.10). Através desta ferramenta, e até ao dia de hoje, é apenas possível exportar os mundos que se encontrem ativos para Windows (os restantes sistemas operativos não suportam ainda realidade virtual). Executando a função “Build” e selecionando a pasta de destino da aplicação, o Unity encarrega-se de criar um ficheiro executável (extensão “.exe”), com uma pasta de dados necessária para que o executável possa ter as informações da aplicação propriamente dita.

De modo a executar a aplicação em qualquer computador, basta colocar o ficheiro executável a correr, e a aplicação funcionará, sendo possível a visualização do modelo biomecânico do parto através dos Oculus Rift e a interação com o Leap Motion.

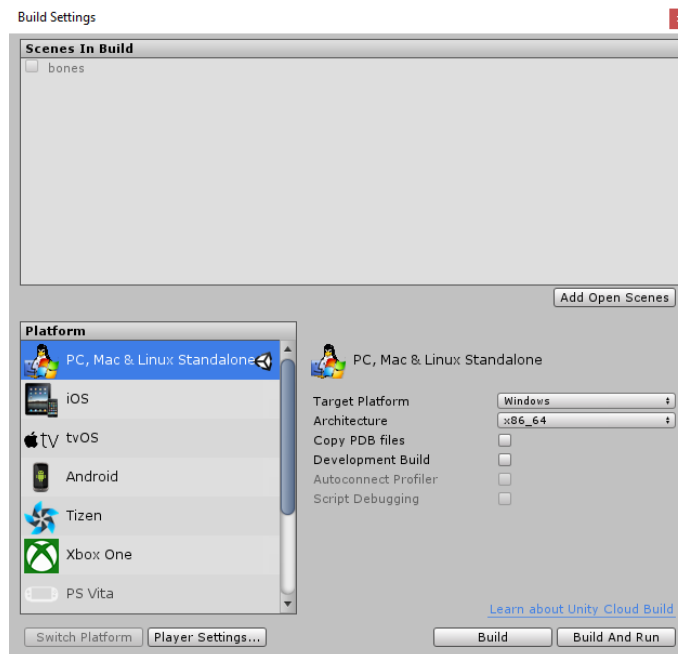


Figura 6.10: Janela de exportação da cena virtual para os vários ambientes permitidos

Capítulo 7

Conclusões e Trabalho Futuro

7.1 Conclusões

O objetivo inicial da presente dissertação consistia na criação de uma aplicação em realidade virtual, capaz de simular um modelo biomecânico de um parto humano através do canal vaginal, modelo este criado através de análises de simulação numérica, funcionando como pós-processamento ao método dos elementos finitos.

Concretizados todos os processos estudados, e entendendo todas as etapas tomadas ao longo da realização do projeto, pode-se concluir que os objetivos propostos foram cumpridos dentro do prazo pretendido.

Houve também momentos durante a realização da presente dissertação em que foi possível efetuar interação com algum público externo ao trabalho desenvolvido, nomeadamente alunos do ensino secundário (no âmbito de atividades decorrentes na Faculdade de Engenharia da Universidade do Porto), bem como durante um congresso doutoral realizado no Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial (INEGI) e durante uma conferência internacional na Universidade do Algarve. Das primeiras interações com o público, retirou-se que o formato utilizado para o menu que executa as funcionalidades descritas anteriormente não é o mais adequado, devido à falta de intuitividade e à incompatibilidade com as diferentes alturas entre os utilizadores, que para pessoas de maior estatura provoca o aparecimento do menu numa posição fora do alcance da visão. Nas restantes interações, esta questão foi prevista, tendo-se executado várias versões do projeto, com diferentes posições para o menu, e o *feedback* dos utilizadores foi bastante positivo, tendo-se obtido como conclusão que o projeto poderá possuir um futuro promissor, e tendo já sido obtida a publicação de um artigo científico numa plataforma digital de artigos de engenharia.

Como forma de resultados finais, foi obtida a aplicação pretendida, que pode agora ser configurada em menos tempo, uma vez que todos os mecanismos de conversão entre os diferentes ficheiros se encontram já completamente entendidos. Além disso, existe um conhecimento mais profundo das ferramentas de realidade virtual, o que origina uma intuitividade superior neste campo que cada vez mais demonstra vir a ser um tópico no quotidiano do ser humano num futuro próximo.

7.2 Trabalho Futuro

Apesar de se terem atingido os objetivos desejados, vários desafios foram encontrados ao longo do desenvolvimento da presente dissertação que colocaram alguns entraves em algumas etapas do projeto e que devem ser reconhecidos neste subcapítulo, alguns deles como futuros progressos a realizar no projeto.

Um dos primeiros encontrado prende-se com o modelo (em formato STL) inicialmente fornecido, no sentido em que este possuía algumas estruturas não desejadas para o caso de estudo. Além disso, alguns constituintes do modelo atravessavam outros componentes (por exemplo, tanto o nariz do feto com uma porção do cóccix atravessam os MPP), algo que não aconteceria numa situação clínica real. O primeiro problema acabou por ser ultrapassado retirando as estruturas não desejadas do modelo, conforme foi descrito no subcapítulo 5.2. Em relação à segunda questão, optou-se por ignorar a não colisão entre as diferentes estruturas, uma vez que o objetivo da presente dissertação não passava pelo manuseamento ou retificação do modelo numérico.

Além disso, novas simulações podem vir a ser incluídas no ambiente virtual, incluindo cenários completamente diferentes da área médica.

Uma questão a resolver em trabalhos futuros seria a implementação de um menu não-estático, que se possa mover de acordo com a posição da palma da mão que executa atualmente a função de ativação ou desativação do menu. Esta funcionalidade permitiria colmatar uma limitação do projeto, no que se refere à adaptação às características específicas dos diferentes utilizadores, nomeadamente a altura de cada pessoa, que causou algumas complicações nos momentos em que foi necessário existir interação com público. De modo a encontrar uma solução temporária para esses momentos de interação, foram criadas várias versões da simulação, com diferentes posições para o menu, para que este pudesse surgir no campo de visão da maior parte dos utilizadores. No entanto, esta solução não foi completamente eficaz, tendo apenas remediado o problema temporariamente. Ao implementar um menu não-estático, seria possível aumentar consideravelmente a intuitividade de utilização da aplicação, assim como a adaptação a qualquer utilizador.

De forma a colocar esta ideia em andamento, foi já investigado e encontrado um projeto semelhante (que até à data se trata de um projeto *open-source*), e que realiza esta mesma ação, possuindo o menu associado ao vetor que o Leap Motion atribui à palma da mão detetada. Este processo começou a ser implementado na presente dissertação, e foi já obtido um objeto não-estático que segue a mão do utilizador, mas devido a restrições temporais e à menor importância deste assunto no objetivo da presente dissertação, esta funcionalidade acabou por não ser incluída na aplicação criada.

Ainda relacionada com o menu, foi ainda pensada uma outra funcionalidade, que consiste no desenvolvimento de novas funções úteis e referias pelos utilizadores aquando das interações com o público. Uma delas seria a existência de um botão de “Reset”, que colocasse o modelo nas condições iniciais da simulação (em termos de posição e orientação, por exemplo). Outro exemplo de função a implementar futuramente seria a escolha de quais os elementos do parto que deveriam ser considerados na escala de cores relativa aos deslocamentos dos nós da simulação,

isto é, deveria ser possível escolher se se pretende visualizar os deslocamentos apenas dos MPP, apenas do feto, ou de ambos, tendo em conta que as simulações para esses casos já existem e foram até referidas na presente dissertação.

Poderiam ainda ser incluídos outros menus que possuísem outras funcionalidades além das já mencionadas, ou que permitissem alternar características do modelo simulado. Por exemplo, poderia ser introduzido uma forma de escolha (de entre várias opções previamente introduzidas) que causasse uma alteração da variável escolhida para análise do modelo (os objetos correspondentes a esta alteração teriam que ter sido incluídos na aplicação) ou que mostrasse opções mais específicas para a opção selecionada.

Em termos da simulação em si, esta pode no futuro incluir uma interação mais personalizada, em particular com o desenvolvimento de controladores eletrónicos específicos para simular cirurgias, o que permitiria a realização e treino de episiotomias, por exemplo. Existem já no mercado simulações semelhantes, mas que são ainda muito pouco precisas e não possuem muitas funcionalidades.

Uma das questões principais do desenvolvimento desta aplicação foi a automatização de todos os processos, de forma a que se pudesse introduzir qualquer modelo biomecânico como *input* (e não apenas o modelo biomecânico do parto utilizado nesta dissertação), e o *output* fosse o ficheiro executável que permitisse a utilização das ferramentas de realidade virtual. Esta característica é possível de se concretizar e foi estudada, tendo sido obtidos os conhecimentos da linguagem de programação Python necessários (linguagem esta que é utilizada por todos os *softwares* referidos, e que permite a utilização dos comandos e ferramentas de cada um deles). No entanto, devido ao elevado tempo que a criação da aplicação consumiu, e devido também a algumas complicações existentes na identificação das estruturas anatómicas nos inúmeros conjuntos de pontos dos ficheiros de texto obtidos (uma vez que a exportação dos ficheiros é realizada de forma desconhecida pelos *softwares* e que foi identificada uma mistura dos pontos pertencentes a diferentes superfícies dos MPP na mesma listagem de pontos), este processo acabou por não ser realizado.

Existe também alguma pesquisa já realizada na área da importação e exportação através do uso de *scripts*, pelo que esse seria um dos primeiros passos a dar no final da presente dissertação.

Por último, terminando a listagem de desafios encontrados e a resolver em trabalhos futuros, encontra-se uma complicação identificada aquando de duas interações com o público, em particular a que ocorreu na Universidade do Algarve, e que se prende com o *hardware* disponível. Para que um computador seja compatível com realidade virtual (em particular com o Oculus Rift), existem algumas especificações mínimas que são necessárias, e que não são muito comuns na maioria dos computadores portáteis. Estas especificações requerem que, para que o Oculus Rift seja utilizado como dispositivo, a placa gráfica seja bastante potente (no mínimo uma NVIDIA GTX 960), que o processador seja bastante poderoso também (no mínimo Intel i3-6100), e que exista alguma memória RAM disponível (8 GigaBytes), entre outras características. Estas especificações existem nos computadores utilizados no local de trabalho, mas não se tratam de aparelhos portáteis, pelo que a movimentação do projeto de modo a que fosse completamente apresentado na conferência internacional não fosse possível. Este problema pode ser considerado como desvantagem

das aplicações de realidade virtual, mas que poderá vir a ser resolvido facilmente com a obtenção de um computador que cumpra os requisitos solicitados, algo que não colocará grandes entraves no futuro, devido à constante evolução das tecnologias.

Anexo A

Ficheiro de Código em C# - FrameByFrameAnimation

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;

public class FrameByFrameAnimation : MonoBehaviour
{
    public bool loop = true;
    public bool play = true;
    public float FPS = 60;
    float frameTime = 0;

    int currentFrame = 0;
    Transform[] frames;
    public Slider sliderAnimation = null, sliderSpeed;

    public int Frame
    {
        get
        {
            return currentFrame;
        }
        set
        {
            if(value >= 0 && value < frames.Length)
            {
                frames[currentFrame].gameObject.SetActive(false);

```

```
        currentFrame = value;
        frames[currentFrame].gameObject.SetActive(true);
    }
}

public void disableSlider(){
    sliderAnimation.interactable = !sliderAnimation.interactable
    ;
    sliderSpeed.interactable = !sliderSpeed.interactable;
}

public void GoBackwards() {
    Frame--;

}

public void SliderPosition(float porcentagem) {
    Frame = (int) (porcentagem * frames.Length);

}

public void SliderSpeed(float porcentagem)
{
    FPS = (int) (porcentagem * 60);

}

public void GoForward()
{
    Frame++;
}

public void Play()
{
    play = !play;
}

void Start()
{
    frames = new Transform[this.transform.childCount];
    for(int i = 0; i < frames.Length; i++)
    {
        frames[i] = this.transform.GetChild(i);
        frames[i].gameObject.SetActive(false);
    }
}
```


Anexo B

Ficheiro de Código em C# - ModelRotation

```
using  UnityEngine;
using Leap.Unity;
using UnityEngine.UI;
using Leap;
using System.Collections;
using System.Collections.Generic;

public class ModelRotation : MonoBehaviour
{
    float speed;
    bool left = false, right = false, up = false, down = false;

    Vector3 meshesCenter;
    FrameByFrameAnimation animationScript;

    Frame frame;
    LeapProvider provider;
    int fingerCount;
    /*****Interface*****/
    public Button play, back, forward, rLeft, rRight, rUp, rDown;
    public GameObject menu;
    /*****Model*****/
    public BoxCollider boxCol;
    float thumbPos, pinkyPos;
    void Start()
    {
```

```
menu.SetActive(false);
speed = 50f;
provider = FindObjectOfType<LeapProvider>() as LeapProvider;
meshesCenter = GetCenterOfMultipleBounds();
animationScript = gameObject.GetComponentInChildren<
    FrameByFrameAnimation>();

boxCol.enabled = true;
}
public void disableUI()
{
    play.enabled = !play.enabled;
    back.enabled = !back.enabled;
    forward.enabled = !forward.enabled;
    rLeft.enabled = !rLeft.enabled;
    rRight.enabled = !rRight.enabled;
    rUp.enabled = !rUp.enabled;
    rDown.enabled = !rDown.enabled;
    animationScript.disableSlider();
    boxCol.enabled = !boxCol.enabled;
}
public void rotateLeft()
{
    left = !left;
}
public void rotateRight()
{
    right = !right;
}

public void rotateUp()
{
    up = !up;
}
public void rotateDown()
{
    down = !down;
}
void Update()
{

```



```
frame = provider.CurrentFrame;

if (frame.Hands.Count > 0)
{
    Hand primeHand = frame.Hands[0];
    foreach (Finger f in primeHand.Fingers)
    {
        if (f.Type == Finger.FingerType.TYPE_THUMB)
        {
            thumbPos = f.StabilizedTipPosition.x;
        }
        if (f.Type == Finger.FingerType.TYPE_PINKY)
        {
            pinkyPos = f.StabilizedTipPosition.x;
        }
    }
    if (primeHand.IsLeft)
    {
        if (thumbPos < pinkyPos)
        {
            boxCol.enabled = false;
            menu.SetActive(true);
        }
        else
        {
            boxCol.enabled = true;
            menu.SetActive(false);
        }
    }
    if (primeHand.IsRight)
    {
        if (thumbPos > pinkyPos)
        {
            boxCol.enabled = false;
            menu.SetActive(true);
        }
        else
        {
            boxCol.enabled = true;
            menu.SetActive(false);
        }
    }
}
```

```

        }
    }
}

if (Input.GetKey(KeyCode.RightArrow) || (right == true &&
    rRight.enabled == true))
{
    gameObject.transform.RotateAround(meshesCenter, Vector3.
        up, speed * Time.deltaTime);
}
else if (Input.GetKey(KeyCode.LeftArrow) || (left == true &&
    rLeft.enabled == true))
{
    gameObject.transform.RotateAround(meshesCenter, Vector3.
        up, -speed * Time.deltaTime);
}
else if (Input.GetKey(KeyCode.UpArrow) || (up == true && rUp
    .enabled == true))
{
    gameObject.transform.RotateAround(meshesCenter, Vector3.
        right, speed * Time.deltaTime);
}
else if (Input.GetKey(KeyCode.DownArrow) || (down == true &&
    rDown.enabled == true))
{
    gameObject.transform.RotateAround(meshesCenter, Vector3.
        right, -speed * Time.deltaTime);
}
}

Vector3 GetCenterOfMultipleBounds()
{
    Bounds bigBound = new Bounds();
    foreach (Renderer r in GetComponentsInChildren<Renderer>())
    {
        bigBound.Encapsulate(r.bounds);
    }
    return bigBound.center;
}
}

```

Referências

- [1] Toda Biologia.Com. Cavidades do corpo humano. URL: http://www.todabiologia.com/anatomia/cavidades_corpo.htm [último acesso em 2017-01-26].
- [2] Rod R. Seeley, Trent D. Stephens, e Philip Tate. *Anatomia & Fisiologia*. Lusociência, sexta edição, 2003.
- [3] F. H. Netter. *Atlas of Human Anatomy*. Saunders/Elvesier, 2006.
- [4] Giancarlo Polesello. Giancarlo polesello - ortopedia especializada em quadril. URL: <http://www.oquadril.com.br/novo/index.php/pelve/> [último acesso em 2017-01-31].
- [5] Ana Rita Monteiro Gomes da Silva. Estudo biomecânico da cavidade pélvica da mulher. Dissertação de mestrado, Faculdade de Engenharia da Universidade do Porto, Julho 2012.
- [6] Peter Williams e Roger Warwick. *Gray's Anatomy*. Churchill Livingstone, trigésima sexta edição, 1980.
- [7] Centrada em Si. O que é o pavimento pélvico e como funciona? URL: <https://centradaemsi.pt/o-que-e-o-pavimento-pelvico-e-como-funciona/> [último acesso em 2017-01-31].
- [8] Slides de anatomia geral ii - universidade unincor. URL: <http://pt.slideshare.net/andreipotting/continuo-slides-anatomia-geral-ii-slide-5> [último acesso em 2017-02-01].
- [9] Gerard Jounghyun Kim. *Designing Virtual Reality Systems - The Structured Approach*. Springer, 2005.
- [10] Gricore C. Burdea e Philippe Coiffet. *Virtual Reality Technology*. Wiley-Interscience, segunda edição, 2003.
- [11] Dispositivos para realidade virtual. URL: <http://www.visgraf.impa.br/sibgrapi97/cursos/rvirtual/disp.htm> [último acesso em 2017-02-01].
- [12] Realidade virtual: Conceitos, evolução, dispositivos e aplicações. URL: <https://periodicos.set.edu.br/index.php/educacao/article/view/909/414> [último acesso em 2017-02-01].
- [13] Dispositivos informáticos utilizados na realidade virtual. URL: <http://api12dmi.blogspot.pt/2012/09/dispositivos-informaticos-utilizados-na.html> [último acesso em 2017-02-03].

- [14] Hp e o monitor 3d. URL: <https://portal.zwame.pt/44374/noticia/hp-revelou-o-zvr-monitor-3d-de-realidade-virtual> [último acesso em 2017-02-03].
- [15] Características da luva de dados. URL: http://marisjoana.blogspot.pt/2013/01/22-realidade-imersiva-e-nao-imersiva_22.html [último acesso em 2017-02-03].
- [16] Inc. Hibbit, Karlsson & Sorensen. *Getting Started with ABAQUS/Explicit - Interactive Version*, 2002.
- [17] Inc. ABAQUS. *Example Problems - Manual*, 2004.
- [18] C. K. Chua, K.F. Leong, e C.S. Lim. *Rapid Prototyping - Principles and Applications*. World Scientific, segunda edição, 2003.
- [19] Ian Gibson, David W. Rosen, e Brent Stucker. *Additive Manufacturing Technologies*. Springer, segunda edição, 2015.
- [20] Mark Pesce. *VRML - Flying Through the Web*. New Riders Publishing, 1996.
- [21] Andrea L. Ames, David R. Nadeau, e John L. Moreland. *The VRML Sourcebook*. John Wiley & Sons, Inc., segunda edição, 1996.
- [22] Eugênio Furstenau. *Novo Dicionário de Termos Técnicos Inglês-Português*. Editora Globo, 1998. Dividido em dois volumes, de A a J e de K a Z.
- [23] Blender foundation. URL: blender.org.
- [24] John M. Blain. *The Complete Guide to Blender Graphics*. CRC Press, primeira edição, 2012.
- [25] Brian R. Kent. Visualizing astronomical data with blender. *Publications of the Astronomical Society of the Pacific*, Junho 2013. Volume 125, pp. 731-748.
- [26] Brian R. Kent. Spherical panoramas for astrophysical data visualization. *Publications of the Astronomical Society of the Pacific*, Abril 2017. Volume 129.
- [27] Brian R. Kent. *3D Scientific Visualization with Blender*. Morgan & Claypool Publishers, 2015.
- [28] A quick tutorial about the fbx ascii format. URL: <https://banexdevblog.wordpress.com/2014/06/23/a-quick-tutorial-about-the-fbx-ascii-format/> [último acesso em 2017-05-26].
- [29] Qingwu Hu, Dengbo Yu, Shaohua Wang, Caiwu Fu, Mingyao Ai, e Wende Wang. Hybrid three-dimensional representation based on panoramic images and three-dimensional models for a virtual museum: Data collection, model, and visualization. *Sage Journals*, Julho 2016. Volume 16, pp. 126-138.
- [30] Ross Tredinnick, Brady Boettcher, Simon Smith, Sam Solovy, e Kevin Ponto. Uni-cave: A unity3d plugin for non-head mounted vr display systems. *Proceedings of the IEEE Virtual Reality Annual International Symposium*, Março 2017. pp. 393-394.

- [31] Qiyun Sun, Wanggen Wan, e Xiaoqing Yu. The simulation of building escape system based on unity3d. *Proceedings of 2016 International Conference on Audio, Language and Image Processing (ICALIP)*, Julho 2016. pp. 156-160.
- [32] Oculus rift. URL: <https://www3.oculus.com/en-us/rift/> [último acesso em 2017-02-03].
- [33] Oculus rift: Dk1 vs dk2. URL: <http://in2gpu.com/2014/08/10/oculus-rift-dk1-vs-dk2/> [último acesso em 2017-02-03].
- [34] Leap motion. URL: www.leapmotion.com.
- [35] Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, e Denis Fisseler. Analysis of the accuracy and robustness of the leap motion controller. *Sensors journal*, Maio 2013. Volume 13, Quinta edição, pp. 6380-6393.
- [36] Jian Cui e Alexei Sourin. Feasibility study on free hand geometric modelling using leap motion in vrml/x3d. *International Conference on Cyberworlds (CW)*, Outubro 2014. pp. 389-392.
- [37] Manuel César Bessa Seixas. Leap motion for wimp based interfaces. Dissertação de mestrado, Faculdade de Engenharia da Universidade do Porto, Julho 2014.
- [38] Leap motion: Api. URL: https://developer.leapmotion.com/documentation/csharp/api/Leap_Classes.html [último acesso em 2017-04-06].
- [39] Leap motion: Gestures. URL: https://developer.leapmotion.com/documentation/v2/javascript/devguide/Leap_Gestures.html [último acesso em 2017-05-30].
- [40] Dulce Alves de Oliveira. *Modeling damage process in pelvic tissues during vaginal delivery*. Tese de doutoramento, Faculdade de Engenharia da Universidade do Porto, Janeiro 2016.
- [41] C.S. Buhimschi, I.A. Buhimschi, A.M. Malinow, G. Saade, R. Garfield, e C. Weiner. The forces of labour. fetal and maternal medicine review. *Cambridge University Press*, Março 2003. Volume 14, pp. 273-307.
- [42] Xinshan Li, Jennifer A. Kruger, Martyn P. Nash, e Poul M.F. Nielsen. Modeling childbirth: elucidating the mechanisms of labor. *WILEY INTERDISCIPLINARY REVIEWS-SYSTEMS BIOLOGY AND MEDICINE*, 2010. Volume 2, Segunda edição, pp. 460-470.
- [43] Heath B. Henninger, Shawn P. Reese, Andrew E. Andersen, e Jeffrey A. Weiss. Validation of computational models in biomechanics. *Proceedings of the Institution of Mechanical Engineers Part H - Journal of Engineering in Medicine*, 2010. Volume 224, Edição H7, pp. 801-812.
- [44] Xavier Fritel, Arnaud Fauconnier, Caroline Levet, e Jean-Louis Bénifla. Stress urinary incontinence 4 years after the first delivery: a retrospective cohort survey. *Acta Obstetrica et Gynecologica Scandinavica*, Outubro 2004. Volume 83, Décima edição, pp. 941-945.
- [45] Montserrat Espuña-Pons, Manuel Fillol, María A. Pascual, Pablo Rebolle, e Ana M. Mora. Pelvic floor symptoms and severity of pelvic floor organs prolapse in women seeking care for pelvic floor problems. *European Journal of Obstetrics & Gynecology and Reproductive Biology*, Junho 2014. Volume 177, pp. 141-145.

- [46] Murat Bozkurt, Ayse E. Yumru, e Levent Sahin. Pelvic floor dysfunction, and effects of pregnancy and mode of delivery on pelvic floor. *Taiwanese Journal of Obstetrics & Gynecology*, Dezembro 2014. Volume 53, Quarta edição, pp. 452-458.
- [47] Julia R. Fielding, Huseyin Dumanli, Andreas G. Schreyer, Shigco Okuda, David T. Gering, Kelly H. Zou, Ren Kikinis, e Ferenc A. Jolesz. Mr-based three-dimensional modeling of the normal pelvic floor in women. *American Journal of Roenthenology*, Março 2000. Volume 174, Terceira edição, pp. 657-660.
- [48] M. Parikh, M. Rasmussen, L. Brubaker, C. Salomon, K. Sakamoto, R. Evenhouse, Z. Ai, e M.S. Damaser. Three dimensional virtual reality model of the normal female pelvic floor. *Annals of Biomedical Engineering*, Fevereiro 2004. Volume 32, Segunda edição, pp. 292-296.
- [49] Y. M. Law e J.R. Fielding. Mri of pelvic floor dysfunction: review. *American Journal of Roentgenology*, Dezembro 2008. Volume 191, Sexta edição, pp. S45-S53, Suplemento S.
- [50] J.A.C. Martins, M.P.M. Pato, E.B. Pires, R.M.N. Jorge, M. Parente, e T. Mascarenhas. Finite element studies of the deformation of the pelvic floor. *Reproductive Biomechanics*, 2007. Série de livros - Annals of the New York Academy of Sciences, Volume 1101, pp. 316-334.
- [51] L. Hoyte, M.S. Damaser, S.K. Warfield, G. Chukkapalli, A. Majundar, D.J. Chei, A. Triverdi, e P. Krysl. Quantity and distribution of levator ani stretch during simulated vaginal childbirth. *American Journal of Obstetrics and Gynecology*, Agosto 2008. Volume 99, Segunda edição, No. do artigo - 198.e1.
- [52] X. Li, J.A. Kruger, M.P. Nash, e P.M.F. Nielsen. Effects of nonlinear muscle elasticity on pelvic floor mechanics during vaginal childbirth. *Journal of Biomechanical Engineering - Transactions of the Asme*, Novembro 2010. Volume 132, Décima primeira edição, No. do artigo - 111010.
- [53] M.P.L. Parente, R.M. Natal Jorge, e T. Mascarenhas. Deformation of the pelvic floor muscles during a vaginal delivery. *International Urogynecology Journal*, Janeiro 2008. Volume 19, Primeira edição, pp. 65-71.
- [54] D. Jing, J.A. Ashton-Miller, e J.O.L. DeLancey. A subject-specific anisotropic viscohyperelastic finite element model of female pelvic floor stress and strain during the second stage of labor. *Journal of Biomechanics*, Fevereiro 2012. Volume 45, pp. 455-460.
- [55] R. Buttin, F. Zara, B. Shariat, T. Redarce, e G. Grange. Biomechanical simulation of the fetal descent without imposed theoretical trajectory. *Computer Methods and Programs in Biomedicine*, Agosto 2013. Volume 111, Segunda edição, pp. 389-401.
- [56] J. Lepage, C. Jayyosi, P. Lecomte-Grosbras, M. Brieu, C. Duriez, M. Cosson, e C. Rubod. Biomechanical pregnant pelvic system model and numerical simulation of childbirth: impact of delivery on the uterosacral ligaments, preliminary results. *International Urogynecology Journal*, Abril 2015. Volume 26, Quarta edição, pp. 497-504.
- [57] S. Janda, F.C. van der Helm, e S.B. de Blok. Measuring morphological parameters of the pelvic floor for finite element modelling purposes. *Journal of Biomechanics*, Junho 2003. Volume 36, Sexta edição, pp. 749-757.

- [58] M.P.L. Parente, R.M. Natal Jorge, e T. Mascarenhas. Computational modeling approach to study the effects of fetal head flexion during vaginal delivery. *American Journal of Obstetrics and Gynecology*, Setembro 2010. Volume 203, Terceira edição, No. do artigo - 217.e1.
- [59] F.G. Cunningham, K.J. Leveno, S.L. Bloom, J. Hauth, D. rouse, e C. Spong. *Williams Obstetrics*. McGraw-Hill Medical, vigésima terceira edição, 2009.
- [60] E. Sorantin, D. Brader, e F.Thimary. Neonatal traumas. *European Journal of Radiology*, Novembro 2006. Volume 60, Segunda edição, pp. 199-207.
- [61] M.H. Beall e M.G. Ross. Clavicle fracture in labor: Risk factors and associated morbidities. *Journal of Perinatology*, Dezembro 2001. Volume 21, No. 8, pp. 513-515.
- [62] R. Eliahou, N. Simanovsky, e N. Hiller. Fracture-separation of the distal femoral epiphysis in a premature neonate. *Journal of Ultrasound in Medicine*, Dezembro 2006. Volume 25, Décima segunda edição, pp. 1603-1605.
- [63] A. Agarwal e I. Verma. Cerebral palsy in children: An overview. *Journal of Clinical Orthopaedics and Trauma*, Dezembro 2012. Volume 3, Segunda edição, pp. 77-81.
- [64] J.E. Muscolino. *Kinesiology: The skeletal system and muscle function*. Elviesier Mosby, segunda edição, 2010.
- [65] K.C. Lien, B. Mooney, J.O.L DeLancey, e J.A. Ashton-Miller. Levator ani muscle stretch induced by simulated vaginal birth. *Obstetrics and Gynecology*, Janeiro 2004. Volume 103, Primeira edição, pp. 31-40.
- [66] H.P. Dietz e V. Lanzarone. Levator trauma after vaginal delivery. *Obstetrics and Gynecology*, Outubro 2005. Volume 106, Quarta edição, pp. 707-712.
- [67] J.A. Ashton-Miller e J.O.L. DeLancey. On the biomechanics of vaginal birth and common sequelae. *Annual Review of Biomedical Engineering*, 2009. Série de livros - Annual Review of Biomedical Engineering, Volume 11, pp. 163-176.
- [68] M.E.T Silva, D.A. Oliveira, e T.H.Roza. Study on the influence of the fetus head molding on the biomechanical behavior of the pelvic floor muscles, during vaginal delivery. *Journal of Biomechanics*, Junho 2015. Volume 48, Nona edição, pp. 1600-1605, Edição especial SI.
- [69] Introdução do leap motion no unity. URL: <https://developer.leapmotion.com/unity#116> [último acesso em 2017-02-09].
- [70] Configuração do leap motion para os oculus rift dk2. URL: <https://developer.leapmotion.com/vr-setup/dk2> [último acesso em 2017-02-09].
- [71] A. I. Kapandji. *Fisiologia Articular - Esquemas comentados de mecânica humana*, volume 1. Maloine, 2000.

